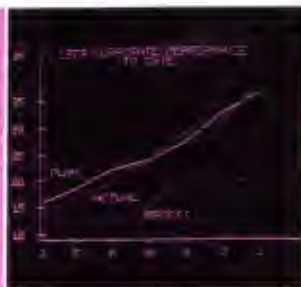




GRAPHICS TABLET™

OPERATION AND REFERENCE MANUAL



<http://www.turbo-2.com/apple/>

NOTICE

Apple Computer Inc. reserves the right to make improvements in the product described in this manual at any time and without notice.

DISCLAIMER OF ALL WARRANTIES AND LIABILITY

APPLE COMPUTER INC. MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS MANUAL OR WITH RESPECT TO THE SOFTWARE DESCRIBED IN THIS MANUAL, ITS QUALITY, PERFORMANCE, MERCHANTABILITY, OR FITNESS FOR ANY PARTICULAR PURPOSE. APPLE COMPUTER INC. SOFTWARE IS SOLD OR LICENSED "AS IS". THE ENTIRE RISK AS TO ITS QUALITY AND PERFORMANCE IS WITH THE BUYER. SHOULD THE PROGRAMS PROVE DEFECTIVE FOLLOWING THEIR PURCHASE, THE BUYER (AND NOT APPLE COMPUTER INC., ITS DISTRIBUTOR, OR ITS RETAILER) ASSUMES THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR, OR CORRECTION AND ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES. IN NO EVENT WILL APPLE COMPUTER INC. BE LIABLE FOR DIRECT, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT IN THE SOFTWARE, EVEN IF APPLE COMPUTER INC. HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF IMPLIED WARRANTIES OR LIABILITY FOR INCIDENTAL OR CONSEQUENTIAL DAMAGES, SO THE ABOVE LIMITATION OR EXCLUSION MAY NOT APPLY TO YOU.

This manual is copyrighted. All rights are reserved. This document may not, in whole or part, be copied, photocopied, reproduced, translated or reduced to any electronic medium or machine readable form without prior consent, in writing, from Apple Computer Inc.

©1979 by APPLE COMPUTER INC.
10260 Bandley Drive
Cupertino, California 95014
(408) 996-1010

All rights reserved.
APPLE product # 030-0076-00

GRAPHICS TABLET™

OPERATION AND REFERENCE MANUAL



TABLE OF CONTENTS

INTRODUCTION

CHAPTER 1 GETTING STARTED

4	What You Will Need
5	Unpacking
6	Plugging In
7	Installing the Interface
9	Backing Up the Diskette
9	Starting Up
10	The Menu Overlay
10	Aligning the Menu

CHAPTER 2 THE GRAPHICS TABLET SOFTWARE

14	Get Ready
14	Drawing
15	The menu
15	Delta
17	The Color menu
18	Some Background Information
18	A Brief Digression on High-Resolution Graphics
19	Guidelines
19	A Clear Alternative
20	Line Up
21	Draw
21	Yes, Sir, Dot's My Baby
22	The Big Frame Up
23	Little Boxes
24	Saving Pictures for Posterity
27	Browsing Through the Catalog
28	Getting Loaded
29	Getting Out
30	Room With a Viewport
33	A Brilliant Reduction

34	Opening the Window
36	Broken Windows
36	Drawing in the Window
37	Reset
38	A Softer Reset
38	Calibrate
40	Mistaken Calibration
40	Long Distance...
41	...And Area Codes
42	Slide Rules
43	Prismatic Apple
44	In Conclusion

CHAPTER 3

PROGRAMMING THE GRAPHICS TABLET

46	The Programs
47	Tablet-Code Applesoft
49	The Main Loop
50	Menu Items
51	Extending the Menu
51	Example: Instant Color Menu
53	Example: Circle Mode
55	Example: Disc Mode
56	The Firmware
57	Tablet Control
60	Accepting Input
61	From Machine Language
64	Quick-Draw
65	By Any Other Name

APPENDIX A

USE AND CARE

68	Care of the Menu Overlay
68	Care of the Tablet
69	Care of the Interface
69	If It Doesn't Work

APPENDIX B

COPYING THE MASTER DISKETTE

72	With Two Disk Drives
72	With One Disk Drive

APPENDIX C

COLOR ANOMALIES

76	Unusual Color Effects...
77	...And How to Get Them

APPENDIX D

PROGRAM LISTINGS

80	Tablet-Code Applesoft
85	Variable Atlas
87	Subroutines
88	Special Locations
89	ROM Code
108	Quick-Draw
116	Utilities

APPENDIX E

SCHEMATIC DIAGRAM

123



INTRODUCTION

Welcome to the world of the Apple Graphics Tablet. The Tablet is a device which converts the position and movements of a special pen into numbers which your Apple can use and understand. The programs, or "software", supplied with your Tablet tell the Apple how to draw pictures on its high-resolution graphics screen, using the information supplied by the Tablet. These programs turn your Apple and Tablet into an artist's sketchpad, an engineer's drawing board, or a mathematician's chalkboard. With the Tablet and the supplied programs you can draw freehand pictures on the Apple's screen, or use the Apple to draw straight lines, rectangular boxes, open frames, or tiny dots. The pictures you create can easily be saved on Apple diskettes and recalled anytime you want. You can use the Apple to calculate the areas and distances of shapes and lines you draw on the Tablet, and you can change the scale of the figures you draw.

This is the Operation and Reference Manual for the Apple Graphics Tablet. The Graphics Tablet is a "hands-on" product, and the best way to learn how to use it is to take pen in hand and start experimenting. Most of this book is based on the assumption that you have the Graphics Tablet set up in front of you, and are following and doing each example as it is presented. If you try to learn how to use the Tablet without using this manual (or even worse, read the manual without actually using the Tablet), you might pick up most of the simpler commands, but you'll never master the more powerful functions of the Tablet. So read the manual, repeat the examples, and don't be afraid to experiment.

The first chapter of this book describes how to set up your Tablet, and what you need in order to use it. Chapter 2 introduces you to the Graphics Tablet software. This is a set of programs which allow you to use the Tablet to draw pictures on the Apple's high-resolution graphics screen. You do not need to know much about the Apple in order to use the Graphics Tablet. In fact, all you really need to know is how to turn it on. Once you start using the Tablet software, it will guide you each step of the way. You do not need to know how to write programs to use the Tablet skillfully and efficiently.

If you do know how to program, you may be interested in Chapter 3. It will give you assistance in modifying the Graphics Tablet software to your liking, including adding your own features to the Tablet menu. There are also instructions on interfacing directly to the Tablet's firmware, so you can write your own special-purpose programs that will use the Tablet. Listings of the programs which operate the Tablet are supplied in Appendix B.

If you see the symbol



it means that the following paragraph contains important information about some Tablet behavior that you might not anticipate. The symbol



means that the following paragraph contains special information you should note. Read these sections carefully.

Above all, feel free to play around with the Tablet. The Apple Graphics Tablet is easy to learn, easy to use, and hard to mess up. With some simple maintenance (described in Appendix A), your Tablet will give you years of enjoyment and use. So sit down at your Apple, take pen in hand, and turn to Chapter 1. We'll let you...

Draw Your
Own
Conclusion!

CHAPTER 1 GETTING STARTED

- 4 What You Will Need
- 5 Unpacking
- 6 Plugging In
- 7 Installing the Interface
- 9 Backing Up the Diskette
- 9 Starting Up
- 10 The Menu Overlay
- 10 Aligning the Menu

WHAT YOU WILL NEED

To use the Apple Graphics Tablet with its supplied software, you will need the following:

- 1) An Apple II or Apple II Plus computer, with 48K bytes of Random Access Memory (RAM);
- 2) If you do not have an Apple II Plus, you will need an Applesoft Firmware card (part number A2B0009), or an Apple Language System (part number A2B0006) with a BASICS language diskette;
- 3) An Apple Disk II plug-in controller card with at least one Disk II disk drive;
- 4) A color or black-and-white video monitor.

In addition, you may wish to have additional Disk II disk drives and controller cards.



The Graphics Tablet was designed to work with most present and future Apple II hardware and software. However, the supplied programs which operate the Graphics Tablet are designed to work with the Apple II DOS disk operating system, versions 3.2 and up. The Graphics Tablet software will not operate under previous versions of DOS or in an Apple Pascal environment.

It is helpful (but not necessary) to have read the following manuals:

- 1) The Applesoft Tutorial (product number A2L0018)
Welcome and Chapter 1
- 2) Do's and Don't's of DOS (product number A2L0012)
Preface through Chapter 2

If you are using the Apple Language System, be sure to read:

- Apple Language System (product number A2L0024)
Chapter 3: Using BASIC

UNPACKING

Your Graphics Tablet package contains ten items:

- 1) The Graphics Tablet and its attached cable.
- 2) The Graphics Tablet's indicator pen and its attached cable.
- 3) A printed-circuit board (the Graphics Tablet Interface card)
- 4) A mylar "menu" overlay.
- 5) Two "GRAPHICS TABLET SOFTWARE" diskettes.
- 6) A piece of die-cut, double-sided foam tape.
- 7) A warranty card.
- 8) A packing list.
- 9) This manual.
- 10) A static cloth.

Save the packing material in case you wish to transport your Tablet -- or in the unlikely event that you must return your Tablet to your dealer for service. If you did not fill out your warranty with your Apple dealer before you brought your Graphics Tablet home, send it in now -- not only does this ensure that any warranty repair your Tablet may need will be done as quickly as possible, but it also puts you on the mailing list for CONTACT, the Apple users' newsletter that keeps you informed of updates and new products.

PLUGGING IN



***** Special Note *****
Before connecting or disconnecting
ANYTHING
on the Apple or
the Graphics Tablet
TURN OFF THE POWER.
This is a must.

Please pay special attention to this warning: If you try to connect or disconnect something from the inside of your Apple when the power is on, there is a good chance that you may damage its electronics.

The Graphics Tablet and its pen connect to the Interface card, which in turn plugs into one of the eight peripheral connector slots in the inside of the Apple, along the back of the main board. The cables attached to the Tablet and the pen terminate in small sockets, which fit over two sets of pins on the Interface card. The sockets are spaced and keyed so that it is very difficult to attach them incorrectly.



Handle the Interface card as you would handle a high-quality, expensive phonograph record. Grasp it only by the corners or edges, and try not to touch the delicate components or pins. Don't grasp the card by the gold "fingers" -- they are the medium through which the Apple communicates to the Tablet and their efficiency is decreased if they are dirty or scratched. The Interface is a precision instrument and should be treated with care.



First attach the Tablet's pen to the interface card. Place the Interface card on a flat surface with the components face up and the gold "fingers" nearest you. Take the connector at the end of the cable from the Tablet's pen. Notice that the four tiny round holes on the bottom of the connector are keyed to correspond to the set of four pins in the upper right corner of the Interface card. Gently slide this connector over the set of pins. There should be some space between the card and the connector. The finished connection should look like this:



Now attach the Graphics Tablet to the Interface card. Take the connector at the end of the cable from the Graphics Tablet. Gently slide the connector over the set of pins near the top middle of the Interface card. When the connector is properly attached there should be some space between it and the card. The finished connection should look like this:



INSTALLING THE INTERFACE

To install the Graphics Tablet Interface card (which you have already connected to the Tablet and its pen) into the Apple, you will simply plug the interface card into the back of the computer, as follows:

1. Turn off the power switch at the back left corner of the Apple. This is important to prevent damage to the computer. Don't unplug the Apple, just turn it off. If you unplug your Apple, you isolate it from the common earth ground and your Apple and Tablet Interface card could be in danger from static discharges.
2. Remove the cover from the Apple. Do this by pulling up on the back edge of your Apple's lid until the corner fasteners pop apart, then slide the lid back and lift it off.
3. Before proceeding, touch your hand to the metal power supply case inside your Apple. This will remove any stray static charges from your hands, so you do not damage the static-sensitive components on the Interface card.
4. Inside the Apple, across the rear of the main green board, are eight long, narrow sockets called Peripheral Connectors, or "slots". The leftmost slot (looking from the keyboard end) is called "Slot #0" and the rightmost is called "Slot #7". The Interface card will operate in any slot except #0, but it is customary for the Tablet to use Slot #5, the third one from the right.
5. Grasp the upper corners of the card between the thumbs and forefingers of both hands. Insert the gold "fingers" of the Interface card into the chosen slot in the back of the Apple, rear edge first. Gently push the front edge of the card down until it is level and firmly seated.

6. Take the two cables which you have connected to the Interface card. On the cable attached to the pen there is a black plastic fitting. This is called a strain relief. There is a hole running lengthwise along the bottom of the strain relief, with a slit running the length of the hole. Pry the slit open with your fingernail and slide the cable from the Tablet through the slit and into the hole. The finished strain relief should look like this:



Now take the strain relief with its "tail" on top and pointing out the back of the Apple and slide it into the leftmost of the two smaller vertical notches in the bottom of the Apple's case. Slide it down to the bottom of the notch. It should be a tight fit. If it doesn't slide all the way down the first time, pull it out and slide it back in again. The plastic is pliable enough so that it will conform to the slot's width after about three or four insertions.



7. Snap the top back onto your Apple. Place the Tablet on a flat surface near your Apple, close enough so that the pen can easily reach all parts of the Tablet surface. Make sure that your disk drive and video monitor are connected properly.

BACKING UP THE DISKETTE

Now that your Graphics Tablet is all hooked up, it's a good time to think about an important rule of thumb. "What rule?" you might ask. The rule is this: Always keep at least one backup copy of any diskette whose information you wish to keep.

The value of a backup copy cannot be overemphasized. Right now, if you were to drop both your Graphics Tablet Software diskettes, and your pet turtle started nibbling on them, or somebody mistook them for square, black Frisbees, or some other catastrophic event occurred which would render them both unreadable, then your Graphics Tablet would be almost useless. Honest. You'd have to write all new programs yourself, or buy another Graphics Tablet Software diskette, in order to use your Tablet.

Take a look at the two Graphics Tablet Software diskettes that came with your Graphics Tablet. Notice that one of them has a small piece of silver tape over the rectangular notch on its edge. This piece of tape is called a write-protect tab. The write-protect tab tells the Apple not to store any more information on the diskette in question. The tab assures that none of the information on the diskette will be accidentally written over. Store this write-protected diskette in a safe place, and use it as your backup copy.

Fortunately, you know better than to leave your Graphics Tablet diskettes lying around where they might be damaged by heat, your pet turtle, or strong magnetic fields. However, you may want to be really careful and keep two backup copies instead of just one. Keeping more than one backup copy insures that your programs will be safe even if one of your backups is accidentally destroyed. If you don't know how to go about making copies of the Graphics Tablet Software diskette, see Appendix B in this manual for instructions.



Don't put your Graphics Tablet Software diskettes, or any other diskettes, on top of the Tablet itself! Its magnetic field will wipe out any information on the diskettes.

STARTING UP

After you've reassembled your Apple and its peripherals and everything is in order, place your Graphics Tablet Software diskette into Drive 1. Remember to use the one that does not have the silver write-protect tab over the rectangular notch on its edge. Now turn the power on and "boot" the diskette. (If you don't understand what this means, STOP! Don't kick your diskette, but read the section called BOOTING DOS in Chapter 2 of your DOS manual, or Chapter 3 in the

Language System Manual if you have an Apple Language System.) The disk drive will whirr and click for about 15 seconds, then the Graphics Tablet logo will be displayed:



To begin your encounter with the Tablet, press the **ESC** key. The screen will display the Graphics Tablet "HELLO Menu", which is a list of things you can do with your Graphics Tablet Software diskette. You'll be using the MENU ALIGNMENT program first.

THE MENU OVERLAY

Included in your Graphics Tablet package is a mylar overlay called the "Graphics Tablet Menu". You will be placing this overlay in the center of the recessed area on the Tablet. The overlay divides the surface of the Tablet into different areas, and each area has a different meaning. Part of the overlay represents the Apple's high-resolution graphics screen, and another part lets you select which functions of the Tablet you want to use.

Once you attach this overlay to the Tablet, you need to tell the Apple the exact location of the overlay on the surface of the Tablet, and the Apple will help you make sure that you've put the overlay on correctly.

ALIGNING THE MENU

Before you use the Tablet, you must first place the overlay on the Tablet and align it. There is a program on your diskette which will assist you in aligning your menu overlay properly. From the Graphics Tablet HELLO menu, press **M** to select the MENU ALIGNMENT program, and then press **RETURN**.

The alignment program tells you what slot your Interface card is plugged into and then creates an information file on your diskette. The name of this file is TAB.INFORMATION. All other programs which use the Tablet can read the vital information about your Tablet and menu from this file. After you run the MENU ALIGNMENT program once, you need not run it again, unless you remove your menu overlay from the Tablet or use your Tablet with a different pen.

The MENU ALIGNMENT program will guide you in attaching and aligning the menu overlay. All you need to do is read its instructions carefully, and do just as it requests. If the menu ever comes loose from the Tablet during the alignment process, press the **[ESC]** key to re-start the whole procedure.

You'll be using four small circles of thin double-sticky foam tape, included with your Tablet, to attach the menu to the Tablet surface. Stick a small circle of tape directly under the target circle in the upper-left corner of the menu overlay, and place the overlay in the center of the recessed area of the Tablet. Stick the overlay to the Tablet surface.

Use the Graphics Tablet's pen to point to the small circle in the upper-left hand corner of the command box labelled RESET. Hold the pen perpendicular to the surface, and carefully press straight down until the point retracts into the pen, making sure that the point of the pen does not slip out of the circle.



Now take the pen and point to the small circle at the lower-left corner of the overlay. Hold the pen straight and press down. The Apple will now determine whether the overlay is straight or crooked. If it is straight, the screen will display "ALIGNED" and you can proceed. If the overlay is crooked, the program will ask you to swing the bottom edge of the overlay a little to one side. Move the overlay just a little in the proper direction and try again. Continue until the screen displays "ALIGNED". Place circles of tape under the remaining three target circles and stick them firmly to the Tablet surface. You may want to press **ESC** to end the program and then re-RUN it to make sure you didn't accidentally move the overlay when you were taping down the corners.

Now follow the arrows displayed on the screen and press the pen down in each small circle in all four corners of the overlay. Be very careful! Make sure that you're holding the pen straight up-and-down, and that the point of the pen does not stray outside of the target. If you do it correctly, you will be rewarded with the message

CREATING TABLET INFORMATION FILE

If you get any other message, you probably slipped somewhere, or the overlay isn't centered on the Tablet surface. Try it again.

Once the overlay is aligned, the Apple will return you to the Graphics Tablet logo. Press **ESC** to get to the program menu again. Now you can start using your Graphics Tablet.

(If you want to be really sure that your menu is properly aligned, you can run the MENU ALIGNMENT program again. Leave the menu taped down and just poke the proper points with the pen. If everything goes well, then your menu is well-aligned. If not, repeat the MENU ALIGNMENT procedure.)

CHAPTER 2 THE GRAPHICS TABLET SOFTWARE

14	Get Ready
14	Drawing
15	The Menu
15	Delta
17	The Color Menu
18	Some Background Information
18	A Brief Digression on High-Resolution Graphics
19	Guidlines
19	A Clear Alternative
20	Line Up
21	Draw
21	Yes, Sir, Dot's My Baby
22	The Big Frame Up
23	Little Boxes
24	Saving Pictures for Posterity
27	Browsing Through the Catalog
28	Getting Loaded
29	Getting Out
30	Room With a Viewport
33	A Brilliant Reduction
34	Opening the Window
36	Broken Windows
36	Drawing in the Window
37	Reset
38	A Softer Reset
38	Calibrate
40	Mistaken Calibration
40	Long Distance...
41	...And Area Codes
42	Slide Rules
43	Prismatic Apple
44	In Conclusion

GET READY

To start using the Graphics Tablet, go to the Graphics Tablet logo, either by re-booting the diskette, completing the MENU ALIGNMENT program, or typing

RUN HELLO

Press **ESC**. Now press **G** to select GRAPHICS TABLET SOFTWARE and press **RETURN**. The disk will whirr and chug for a while, and the Apple will present you with a blank screen. (If you get a message informing you that the Tablet information file does not exist, press [RETURN] and run the MENU ALIGNMENT program.) In about three seconds, your Tablet will be ready to use.

DRAWING

Touch the point of the Tablet's pen lightly to the surface of the Tablet. Move the pen around. You should see a small "crosshairs" cursor moving around the screen as you slide the pen around. The crosshairs are a locator, and the position and motions of the crosshairs on the screen correspond to the position and motions of the pen on the Tablet. Now press down on the pen so that the point retracts, and start drawing. As you draw on the Tablet, the path you trace will show up on the screen as a thin white line.



The top and sides of the working area of the overlay (the area with the fine mesh gridwork) correspond to the top and two sides of the Apple's screen. However, the working area on the overlay is slightly taller than the screen. To compensate for this difference in height, only the upper 2/3 of the overlay's working area is "mapped onto" the screen. The rest, about 2.5 inches (6.35 cm) at the bottom of the working area, is not usually active. (For information on how to use

the full working area, see the WINDOW command.) You might want to find the lower boundary of the working area and mark it with a felt tipped pen on the overlay.

THE MENU

Along the top of the Tablet's Mylar overlay are two rows of 22 squares. Each square in the top row carries the name of a certain command or function which the Tablet software can perform. These two rows of squares are called the Tablet Menu. They let you order functions for the Tablet as you would order food in a restaurant in a foreign country: by pointing to what you want. If you could speak the proper language, you would order dinner by telling the waiter what you would like. But the Tablet's language consists of thousands of magnetic and electrical impulses traveling near the speed of light. Most people can't communicate in this fashion (those who can are mutants, and thus have gone far in the computer world), so you'll have to indicate your choices to the Tablet by pointing at the Menu.

To invoke a command or function, touch the point of the Tablet's pen anywhere inside the corresponding square and press down. Hold the pen down until you hear the Apple beep. If you don't hear a beep, then you haven't fully activated the command, and you should lift the pen and try again.

The second row of boxes, which carry no name, consequently have no function. You can use them for your own programs (see EXTENDING THE MENU in Chapter 3).

The following pages describe each command and its function. To help you locate the square for each command, the section describing that command will be headed with a drawing of the menu and a pen pointing to the proper square.

DELTA

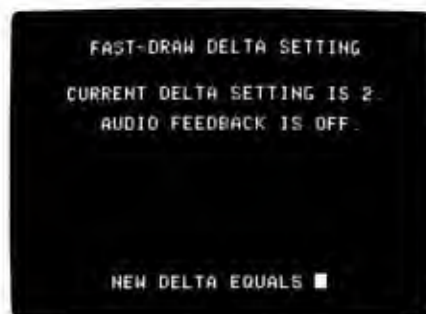
RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	-----------	---------	-----------	------

The DELTA function lets you adjust the precision with which the pen draws on the screen. The Apple subdivides the working area of the Tablet into 53,760 small dots, each one corresponding to one dot on the Apple's screen. As you move the pen around the surface of the Tablet, the Apple draws lines between the dots you traverse. The DELTA setting lets you control the distance the pen can move between the Apple draws a line to the new dot. The smallest possible DELTA is 1. This setting will make the Apple draw a new line each time you move the pen a vertical or horizontal distance of one dot from the

Tablet, about .039 inches or 0.997 mm) from the last dot plotted. The normal value for DELTA is 2. The largest DELTA value is 127. This will make the Tablet draw a new line only after the pen has moved a horizontal or vertical distance of 127 dots (4.98 inches, or 12.6 cm) from the last dot plotted.

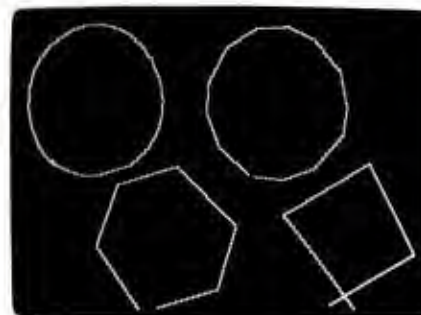
Associated with the DELTA setting is the Audio Feedback feature. When this feature is turned on, the Apple's speaker will emit a click each time the Apple draws a new line. With the Audio Feedback feature enabled, you can actually hear as well as see the effects of different DELTA settings.

To look at or change the current DELTA setting, touch the pen to the DELTA command square. Press it down until you hear the Apple beep. You'll see the following:



The first few lines tell you the current DELTA setting and whether the Audio Feedback feature is ON or OFF. The Apple will ask you for the new DELTA value. If you wish to retain the current DELTA value, just press **RETURN**. If not, type a number between 1 and 127 and press **RETURN**. Next, you'll be asked whether you want the Audio Feedback ON or OFF. Again, if you wish to retain the current setting, just press **RETURN**. Otherwise, type the word ON (to produce the clicks) or OFF (to silence the clicks) and press **RETURN**. The Apple will then return you to the picture you were drawing, with the new DELTA and Audio Feedback settings in effect.

Experiment a little with different DELTA settings. Set a cup or saucer on the Tablet surface and trace its perimeter several times, using different values for DELTA each time. You'll get something which looks like this:



Turn the Audio Feedback ON and OFF, and use it at different DELTA settings. At low settings, it will buzz as you move the pen around; at higher settings, you'll be able to detect distinct clicks.

THE COLOR MENU



The Apple's screen can display six colors: black, white, green, violet, orange, and blue (of course, if you are using a black-and-white monitor, you'll see only various shades of grey). The Apple lets you draw on the screen with all of these colors.

Touch the pen to the command box marked PEN COLOR and press down. The Apple will beep, the screen will clear and the message

CONSTRUCTING COLOR MENU

will appear at the bottom of the screen. The Apple will proceed to draw eight colored boxes, surrounded by a grey border.

Move the pen lightly across the surface of the Tablet. You'll see a small block drifting around the screen (instead of the usual crosshairs). Use the pen to position the block over the color with which you wish to draw, and press down. The color menu will vanish, and you will be looking at the screen on which you were previously drawing. Now, draw! The lines you draw will be in the color you selected. Change colors again and keep drawing. All the rules are the same. Only the colors have been changed.

If, while you're shopping around for a new PEN COLOR, you decide you really don't want to change the color you've got, just press **RETURN**. Your PEN COLOR will not be changed.

SOME BACKGROUND INFORMATION

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALL BRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	-------------	-------	---------------	--------------	---------------	---------	--------------	------

When you start drawing with the Graphics Tablet, you're given a black screen on which to create. You can tell the Tablet that you wish to use a different-colored background by pressing the pen in the BG COLOR (BackGround COLOR) square. The Apple will present you with a color menu (as for the PEN COLOR command). Pick the color you want to use as a background; for instance, orange. The menu will vanish and the screen will instantly be filled with orange, or whatever color you have chosen.



Using the BG COLOR command will erase everything you had on the screen, so if you want to specify a BackGround COLOR, do it before you start to draw.

Are you trying the examples? Is the BackGround COLOR command working? Is orange your favorite color? Again, if you decide not to change the BackGround COLOR, just press **RETURN** instead of selecting a color. Your BackGround COLOR (and your picture, too) will be left unchanged.

A BRIEF DIGRESSION ON HIGH-RESOLUTION GRAPHICS

by now you must have noticed that there are some funny things going on with the colors. For example, set the BackGround COLOR to green and try to DRAW blue lines across it. Or set the BackGround COLOR to violet, and draw some blue lines. Obviously there's something wrong. The color "shadows" and the "zebra stripes" which you see on a color television set, or the strange distortions, unevenness, and lack of consistency you observe on a black-and-white monitor, are the results of the Apple's method of generating colors in its high-resolution graphics display. For more information on the anomalies of the Apple's high-resolution graphics color generation scheme, see Appendix C.

GUIDELINES

To minimize the problems created by the Apple's high-resolution graphics color scheme, follow these guidelines:

- 1) Most inconsistencies of the Graphics Tablet colors occur with vertical lines. Use horizontal lines when possible.
- 2) When you're drawing with black or white on a colored background, or in color on a black or white background, draw the lines a little thicker than normal by going over them twice. This takes care of the broken lines you may get.
- 3) If you need to place two colored blocks next to each other, stack them vertically, not horizontally. This cures the colored shadows that sometimes appear between colors.

So much for the digression, on with the Tablet.

A CLEAR ALTERNATIVE

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALL BRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	-------------	-------	---------------	--------------	---------------	---------	--------------	------

If you're tired of the scribbles and doodles on your screen, press the pen to the CLEAR square. Zap! Your whole screen will be restored to the BackGround COLOR (see the previous section). Draw mode will be restored, and, if you haven't set one, the BackGround COLOR will be black.

If you have set a VIEWPORT (described a little further on in this chapter), then CLEAR will affect only the portion of the screen inside the VIEWPORT. The rest of the screen will remain unchanged.

LINE UP

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

So far, you've been happily drawing somewhat rough, freehand lines on the Apple's screen. If you wanted to draw a straight line between two points, you probably tried to draw it with a straightedge (smart, but awkward) or did it freehand (sloppy). "Is there a better way to draw straight lines?!", I hear you cry. Well, guess what! Yes, there's a better way to draw straight lines. Press the pen to the box which, for some obscure reason, bears the designation LINES. Now you have entered LINES mode. You will remain in LINES mode until you tell the Apple otherwise. We'll tell you how to do that later.

Meanwhile, since you're in LINES mode, let's draw some lines. Press the pen down anywhere on the Tablet's working area and lift the pen again. See the small dot left on the screen? That will be one endpoint of your line. Now press the pen down at another point in the working area. Zap! There's now a straight line connecting the two points. Press the pen down again at another point, and the Apple will draw another line, this one connecting the new point and the second point. Now rush to your nearest toy store and buy a Connect-the-Dots coloring book. Pick out an interesting page, tape it to your Tablet, and start connecting dots. The figure will magically appear on your screen.

If you want to start a second LINES figure, simply press the pen to the LINES command box again. The next point at which you press the pen will be the beginning of a new figure.



The "straight" lines you draw with your Tablet may not seem absolutely straight to you. This is normal. Lines that are neither horizontal nor vertical are actually made up of tiny zig-zags between dots on the screen.

Once you enter LINES mode, you'll stay in LINES mode until you ask to leave. The proper way to ask to be excused is to press the pen to a box that represents another drawing mode.

DRAW

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	-----------	---------	-----------	------

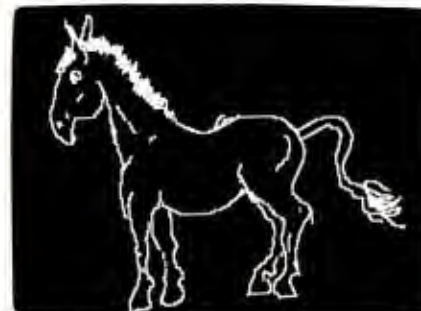
The mode you were in before you entered LINES mode is called DRAW mode. DRAW mode is the normal state of the Graphics Tablet and is automatically put into effect when you choose the Graphics Tablet Software from the diskette menu. This means that DRAW mode is the default mode.

Whenever you wish to leave a fancy drawing mode (LINES, BOX, FRAME or DOTS), simply press the pen to the command square called DRAW. Your picture will be left intact and you will be able to draw normally until you specify another mode.

YES, SIR, DOT'S MY BABY

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

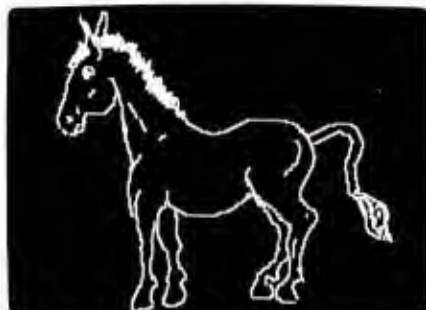
Once you've got a picture on the screen, you might want to edit or change small portions of it. For example, you've drawn this picture:



and you want to fix up the little "glitches" around the edge. There are a couple of ways to do this: you could set the PEN COLOR to black and DRAW the glitches out, you could erase whole portions of the screen and redraw them, or you could simply erase the whole thing and start over. Fortunately, there's an easier way. Press the pen to the square marked DOTS. You are now in DOTS mode, and will remain in DOTS mode until you specify another. While you are drawing with DOTS, the

Graphics Tablet will let you plot individual points on the screen. Each time you press the pen down in the working area you will plot one, and only one, point on the screen. When you lift the pen up again and press it down in a new place, you will plot another single dot. The dots will be of the color you specified in the most recent **PEN COLOR** command, or white if you have not selected any other color.

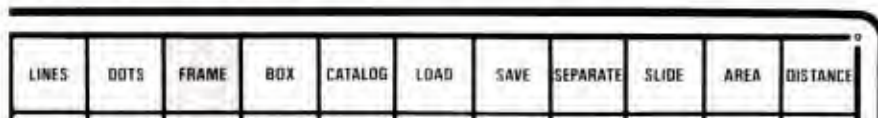
By setting the **PEN COLOR** to the **BackGround COLOR** (normally black), setting **DOTS** mode, centering the crosshairs on the extraneous glitches in the picture, and exercising them one by one, you can turn a rough picture like the previous one into this:



DOTS mode is also handy for adding shading and texture to your pictures. **DOTS** mode is most useful when used with **VIEWPORT** and **REDUCER**, described later in this chapter.

To leave **DOTS** mode, press the pen in the command square for any other mode (like **DRAW**, **LINES**, **BOX** or **FRAME**).

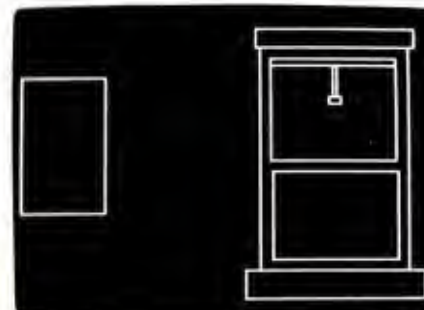
THE BIG FRAME-UP



When you're in the **FRAME** mode you can draw open rectangular boxes on the screen by specifying two diagonally opposite corner points. To enter **FRAME** mode, press the pen down in the (surprise!) **FRAME** command box. Now press the pen down anywhere on the Tablet's working area and lift it again. A single dot will appear on your screen. Take the pen and press it down at another point on the working area. The Apple will draw an open rectangle with opposite corners at the points you specified. Pick another point and press the pen down. Notice that the **FRAME** mode doesn't draw a frame with the new point and previous

point (as **LINE** mode would draw a line between them), but instead uses the new point as a corner of a separate **FRAME**. Pick and press a fourth point to complete the second **FRAME**.

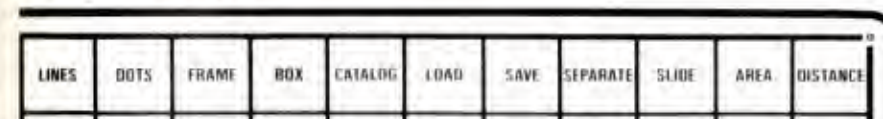
Your **FRAMEs** can be simple, or you can use many **FRAMEs** to make a larger, more complicated **FRAME**:



You can draw your **FRAMEs** in different colors, too. The **FRAMEs** will be drawn in the current **PEN COLOR**, or white if you haven't selected any other color. Beware! Colored **FRAMEs** may come out with a side or two missing because of the nature of the Apple's high-resolution graphics screen (see Appendix C). If this happens, re-draw the **FRAME**, but move the corner points very slightly to one side.

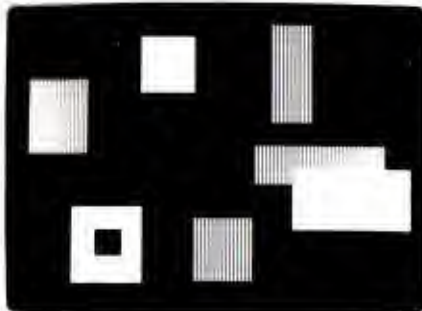
To leave **FRAME** mode, press the pen in the square for any other mode (such as **DRAW**, **LINES**, **DOTS**, or **BOX**).

LITTLE BOXES



There's a white one, and a blue one,
And a green one, and an orange one,
And they're all made
On the Graphics Tablet
And they all look
Just the same.

Now **FRAMEs** are nice, but they're kind of vapid. You might even go so far as to say they're empty. If you're looking for something a little more, well, fulfilling than an ordinary rectangular quadrilateral, then the **BOX** mode is for you. Press the pen down in the square marked **BOX**. Now press the pen down at two points on the working area, as you did for **FRAME**. The Tablet software will give you solid indication that the task is completed by drawing a uniform, monolithic box with corners at the two points you specified.



You will remain BOXed into this mode until you free yourself by pressing the pen down in one of the squares marked DRAW, LINES, DOTS, or FRAME. The BOXes you draw will be of the current PEN COLOR. If you have not specified a different color, your boxes will be white.

SAVING PICTURES FOR POSTERITY

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

By now you should have generated some beautiful (well, at least interesting) artwork. It's a shame that you have to erase it, isn't it? Well, you can save the entire picture for later recovery and further work by pressing the pen in the square marked SAVE. Your picture will vanish (temporarily) and the screen will display the message

TYPE A NAME FOR THIS PICTURE.

==>

A picture name can be from 1 to 26 characters long, and may include letters, numbers, and special characters (except the comma). Unlike normal diskette file names, picture names do not need to begin with a letter; you can have picture names such as

1 FOR THE ROAD

or

<<SPACE>>

(notice the spaces before the name)

The reason for this is that before the Apple saves the screen onto the diskette, it adds the prefix "PIC." to your picture name to identify it as a bona fide Graphics Tablet Picture. Since diskette file names will always begin with the letter P (in PIC.), your picture names can begin with whatever you please.

The PIC. flag also implies that the picture file includes the Tablet WINDOW setting (see the WINDOW command). Files which do not contain this information should not carry the PIC. flag.

After you type the name of the picture, press **RETURN**. If you decide you don't really want to save the picture yet, just press **RETURN** without typing any name. Your picture will reappear, and you'll be left in DRAW mode.

If you do choose to save your picture, the Apple will then ask you:

DRIVE? ==> (DEFAULT=1)

The Apple will save your picture onto the diskette in the disk drive you indicate. The DEFAULT drive is the drive which the Apple thinks you'll want to use, drive 1 the first time and the drive specified previously each time thereafter. Type the drive number you wish to use and press **RETURN**, or just press **RETURN** to select the DEFAULT drive. (If you try to specify any drive number other than 1 or 2, the Apple will use the default drive). The drive will whirr and chug for a moment, then your freshly saved picture will reappear, in DRAW mode with PEN COLOR as it was when you left.

If there is already a picture on the selected diskette with the name you specified, the Apple will display the message

A PICTURE ALREADY EXISTS WITH THAT NAME.

CONTINUE (Y OR N)

If you wish to overwrite the current picture which has the name you specified, press **Y** **RETURN**. If you don't want to destroy the picture on the diskette, press **N** **RETURN** and repeat the SAVE operation using a different picture name (a lone **RETURN** is accepted as an **N** **RETURN** response).

If you complete the SAVE procedure, or if your attempt to SAVE a picture is foiled, and you get an error message from the Apple, you will lose any VIEWPORT you may have set (see the VIEWPORT command). If, however, you have aborted the SAVE command with an **N** **RETURN**, the VIEWPORT will remain intact.

If you receive this message:



then any number of things could be wrong: the diskette is full and can hold no more pictures, the diskette is write-protected, or there's another picture on the diskette with the same name and the file which holds that picture is locked. In the first case, simply use another uninitialized diskette. In the second case, remove the diskette, peel off the write protect tab and reinsert the diskette. In the third case, try another file name. Whatever the problem is, you may press the spacebar to attempt to SAVE the picture again under the same name, or press **RETURN** to cancel the attempted SAVE. Your picture will reappear, and you will be back in DRAW mode.



If you filled up the diskette by trying to SAVE a picture, only part of the picture will actually be stored on the diskette. It is best to delete the partial file from the diskette after you have SAVED the picture on another diskette (see GETTING OUT).

If you receive this message:



while attempting to SAVE a picture, then there are problems. Maybe you specified Drive 2 when you only have one drive, or the diskette is

uninitialized, or the data on it has been destroyed. You could have a faulty disk drive or controller card, or your drive is under the influence of a powerful magnetic field (did you put it on top of your television? Naughty, naughty). There may be no diskette in the drive, or you left the drive door open, or the diskette is crimped and is not rotating. Whatever the cause, press the spacebar to attempt the SAVE again, or press **RETURN** to get back to your picture. Investigate.

BROWSING THROUGH THE CATALOG

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

You can look at the contents of your diskette by pressing the pen in the square marked CATALOG. The Apple will ask you (as above) for the drive number. You can select the DEFAULT drive by pressing **RETURN**.

The message

PRESS SPACE BAR TO CONTINUE

will be centered at the top of the screen. The drive will whirr a bit, and the names of all files on the diskette will be presented. Don't press the spacebar yet! The file names which begin in PIC. are your Graphics Tablet pictures, and should all have the annotation B 033 to the left of them. Incidentally, the B denotes that they're BINARY files, and the 033 means that they use 33 diskette sectors, or a little over 8K bytes of memory, each. If you see any PIC. files which aren't marked B 033, then they're not complete pictures. Change their names (see GETTING OUT) so you don't mistake them for Graphics Tablet pictures in the future.

If you order a CATALOG of the Graphics Tablet Software diskette, these files will be included:



These are all component programs of the Graphics Tablet package. The only notable file is the GRAPHICS TABLET LOGO, which you'll notice has the notation B 014 to the left of it. (The 034 means that it's

slightly larger than normal Tablet pictures, which are labeled B033. This causes no problems, however.) This is the picture of the Graphics Tablet Logo frame, which you see when you boot the diskette. You can LOAD this picture and work on it, even though it's not a PIC. file (see LOAD, below, for details).

CATALOG can fall victim to the same I/O ERROR problems as noted in SAVE. See the previous section for details.

If the CATALOG listing is too long for the screen, the listing will pause after displaying 18 files. Press the spacebar to get the rest of the CATALOG.

When you've finished looking at the CATALOG, just press the spacebar. Your picture will instantly reappear on the screen, with PEN COLOR unchanged.

GETTING LOADED

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE

Once you have SAVED a picture on diskette, you can call it back to your screen to be worked on some more, or just bring it out so you can admire it for a minute. Press the pen to the square marked LOAD. The following words will appear:

PLEASE TYPE THE PICTURE NAME.
==>

If you change your mind and don't want to LOAD a new picture, press **RETURN**. Your previous picture will reappear, and you will be left in DRAW mode with the same PEN COLOR as when you left.

If you do want to LOAD another picture, type the name of the picture which you wish to see, and press **RETURN**. You don't have to type the PIC., the Apple will supply that for you. The rules for naming pictures are the same as described in the SAVE command.

You will then be asked to specify which drive the diskette with the chosen picture is in. Press **RETURN** to indicate that it's in the DEFAULT drive, or type the drive number (1 or 2) and press **RETURN**.

The disk drive will spin for a few moments, then the selected picture will appear on the screen. You will be left in DRAW mode.

LOAD is vulnerable to the same disk I/O ERROR problems as were described in the SAVE section.

You can LOAD picture files which were not created by the SAVE command, as long as they carry the notation B 034 in their CATALOG listing. One such file is the GRAPHICS TABLET LOGO file on your Graphics Tablet Software diskette. Even though this file doesn't have the PIC. flag in front of its name, it can be LOADED and worked upon like any other picture. If you LOAD this file, and SAVE it again, the new version will have the prefix PIC. attached to the name, and will have the notation B033 to its left in the CATALOG.

When the Apple sees the PIC. prefix, it infers that the file contains information about the Tablet WINDOW setting along with the picture. The absence of the PIC. flag indicates to the Apple that it should use the default WINDOW setting (see the WINDOW command). In addition, if you have a picture on the diskette whose file name does have the PIC. prefix, you can make the Apple ignore the Tablet WINDOW setting in that file by typing the PIC. prefix at the beginning of the file name when you LOAD it.



If you LOAD a picture which was SAVED on another Apple or Graphics Tablet, it's possible that the Tablet which created that picture uses a slightly different WINDOW setting than yours. The difference usually appears as a discrepancy between the motions of the pen across the working area and of the crosshairs on the screen. If the crosshairs don't correspond to the pen position, then re-LOAD the picture, but type the PIC. prefix at the beginning of the file name. This will make the Apple use the proper WINDOW setting for your Tablet.

With one exception, an attempt to LOAD a picture, whether successful or not, will remove any VIEWPORT you may have set (see the VIEWPORT command). If you have aborted the attempted LOAD with a **N** **RETURN** or a **RETURN**, this rule does not apply.

GETTING OUT

With the Graphics Tablet, you can perform three simple operations with disk files: SAVE, CATALOG, and LOAD. In order to RENAME or DELETE picture files, you'll have to leave the Graphics Tablet Software and get back to the Applesoft/DOS command level. To do this, get to DRAW mode and press **ESC**. The Apple will ask you if you indeed wish to leave.



If you answer **Y**, you will lose whatever picture you had on the screen! Any other reply will send you back to your artwork, in DRAW mode.

If you answer **Y**, then the Apple will run the HELLO program on the diskette, and you will see the Graphics Tablet Logo frame (see Chapter 1, STARTING UP).

Press **ESC** again to get to the HELLO menu. Choose **Q** to QUIT and press **RETURN**. The screen will be cleared and the Applesoft prompt character (I) will appear in the upper-left corner.

Now you can DELETE, RENAME, LOCK, UNLOCK, or VERIFY any of your picture files on the diskette, or do almost anything else in Applesoft or with DOS. (For details on how to perform these operations, see Chapters 2 and 4 of your DOS manual.) Remember to include the PIC. at the beginning of the pictures' file names! To return to the Graphics Tablet software, type

RUN HELLO

When the Graphics Tablet logo appears, press **ESC**, select **G** for Graphics Tablet Software, and press **RETURN**. You'll be working with the Tablet again, with a blank screen, a white pen, no VIEWPORT, the WINDOW at its default setting, the REDUCER off, and in DRAW mode.



DON'T try to RUN the file GRAPHICS TABLET SOFTWARE directly! It is not a program in itself, but is an EXEC file which runs several programs and sets up some parameters necessary for the well-being of the Tablet software. You should always enter the Graphics Tablet Software by selecting it from the HELLO menu.

ROOM WITH A VIEWPORT

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALI-BRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	------------	---------	-----------	------

Am I to spend the rest of my short life
Confined by these four corners, bright and sharp?
Shall I be limited in my designs
To draw only within this VIEWPORT small?

This cannot be! And yet, there's recompense:
The box which limits, also can protect,
And keep me from destroying what I've wrought.
Confine, protect; the VIEWPORT functions thus.

You can use the VIEWPORT command to select a rectangular area on the screen. Once you set an area for a VIEWPORT, you will be allowed to draw only within that area. This allows you to concentrate on one

area of the screen at a time, while protecting the rest of the picture from being accidentally erased or overdrawn.

The VIEWPORT appears on the screen as four small "L"-shaped corner marks, one at each corner of the VIEWPORT. Each leg of each "L" is three dots long and one dot high. The VIEWPORT itself is the area enclosed by these four corners (the corners are actually outside the VIEWPORT proper). If you change or remove a VIEWPORT, the corner marks vanish without a trace, leaving the screen under them unchanged.

To specify a VIEWPORT, press the pen to the VIEWPORT square on the Tablet menu. The prompting message

UPPER LEFT?

will appear briefly at the bottom of the screen. Position the pen at the spot where the upper-left corner of the VIEWPORT should be (imagine you're drawing a FRAME) and press down. One corner mark will appear, and another prompting message:

LOWER RIGHT?

will be displayed:



Position the pen at the opposite (lower-right) corner of your proposed VIEWPORT, and press down. The other three corners of the viewport will appear. This is how the finished VIEWPORT will look:



Unlike FRAME and BOX modes, in which you can specify the corner points in any order, VIEWPORT really does want the second corner point to be below and to the right of the first. If you give the points in reverse order, or specify an impossible VIEWPORT (one which has no height or width), then you will receive the message

PLEASE SPECIFY POINTS CORRECTLY

You will then be asked for both corner points again.

Once you've set a VIEWPORT, what do you do with it? Simple, just DRAW. After you specify the two corner points, and you see the four-cornered frame, you will be placed in DRAW mode with the PEN COLOR unchanged. Anything you DRAW outside of the VIEWPORT simply will not show up on the screen; anything you draw inside it, will. Notice that the crosshairs will appear even outside the VIEWPORT, but pressing down on the pen has no effect.

You can change to any other drawing mode (BOX, DOTS, LINES, or FRAME) and it will work normally inside the VIEWPORT. But if, while you're in one of these modes, you try to specify a point outside the VIEWPORT, you'll receive the admonition

POINT OUTSIDE VIEWPORT. RESPECIFY.

Just choose another point inside the VIEWPORT. If you want to restart the BOX, FRAME, or LINE you're drawing, press the pen to the square for the proper mode again.

If you invoke the VIEWPORT command when another VIEWPORT is already active, the Apple will remove the previous VIEWPORT before asking you to specify a new one.

At any time after you have invoked the VIEWPORT command, but before you have finished specifying a new VIEWPORT, you can tell the Apple to give you one of two special VIEWPORTs. One of these is the VIEWPORT you were using before you started to set a new one, and the other is the "default" VIEWPORT (the currently set WINDOW).

To recover the VIEWPORT you had before you started to set a new one, press **RETURN** before you finish the VIEWPORT command.

To request the default VIEWPORT, press **0** before you finish the VIEWPORT command. The default VIEWPORT is the full screen, or (if you have invoked the WINDOW command) the area within the WINDOW. When the VIEWPORT is set to the full screen, no corner marks appear.

The VIEWPORT command always leaves you in DRAW mode with the PEN COLOR unchanged.

A BRILLIANT REDUCTION

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CAL- BRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	-------------	-------	---------------	--------------	---------------	---------	--------------	------

Once you've set a VIEWPORT, you can use the REDUCER function to shrink the entire Tablet working area into the VIEWPORT on the screen. This allows you to convert large pen motions on the Tablet into small motions on the screen. This lets you make precise, small drawings. When you use the REDUCER in conjunction with the DOTS mode, you can modify very small areas of a picture, setting and resetting individual dots if necessary.

Once the REDUCER is enabled, it will stay in effect until you remove it or change the VIEWPORT. To use the REDUCER, set a VIEWPORT around the area in which you wish to work, then press the pen to the square marked REDUCER. When you hear the Apple beep, the REDUCER is active. If you receive the message

NOT POSSIBLE.

then you have specified a VIEWPORT which is too small or too disproportionately shaped for the REDUCER to function. Such an impossible reduction will leave you with the REDUCER inactive and everything else unchanged.



The screen position of the VIEWPORT determines the minimum possible size into which you can REDUCE. You can REDUCE into smaller VIEWPORTs in the upper-left corner of the screen than in the lower-right corner. Specifically, the smallest possible VIEWPORT size into which you can REDUCE ranges from two screen dots square (at the normal WINDOW setting) in the upper-left corner to 45 dots square in the lower-right.

To disable the REDUCER, press the pen to the REDUCER square again. The Apple will beep and the REDUCER will be disabled. The RESET, SOFT RESET, WINDOW, VIEWPORT, and LOAD commands also disable the REDUCER.

When the VIEWPORT is at its default setting, the REDUCER has no effect.

OPENING THE WINDOW

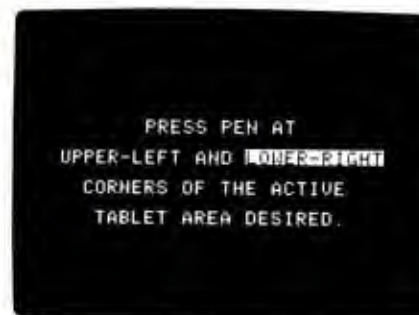


The WINDOW command works like a VIEWPORT with the REDUCER on, but the other way 'round. Where VIEWPORT with REDUCER lets you draw something large on the Tablet, and have it appear smaller and in a specific place on the screen, the WINDOW lets you draw something small in a specific place on the Tablet and have it appear large on the screen.

You normally set a WINDOW before you begin drawing a picture. Press the pen to CLEAR and then to WINDOW. This will appear on the screen:



Find a picture of a molehill and tape it to the Tablet's working area. Now take a pen or a pencil (not the Tablet's pen!) and draw a box around the significant part of the picture. Take the Tablet's pen and, following the highlighted instructions on your screen, press it to the upper-left corner of the box. The highlighting will shift:



Press the pen to the lower-right corner. The words will disappear and the drawing screen will return, with a large frame in the middle of the screen. This frame is proportional to and corresponds with the frame around the molehill on the Tablet, and is centered on the screen. Take the pen, set DRAW mode, and trace the molehill. You will make a mountain on your screen out of the molehill taped to the Tablet.



After it's finished, WINDOW returns you to DRAW mode, with your PEN COLOR unchanged and the VIEWPORT set to the same size as the WINDOW frame on the screen.

The reason that you aren't shown the screen and crosshairs while you set the WINDOW (as you are when you set a VIEWPORT) is that you're selecting an area on the Tablet, not the screen. The resulting area on the screen is as large as the Apple can make it, proportional in size to the WINDOW on the tablet, and centered on the screen. Since the WINDOW area on the Tablet bears little relation to the screen before it's set, the screen and crosshairs are not displayed.

At any time after you have initiated the WINDOW command and before you have completed it, you can use the Apple's keyboard to indicate that you want the default WINDOW, (the entire working area of the Tablet.)

or that you want to cancel the WINDOW sequence. Press **RETURN** at any time during the WINDOW sequence to cancel it; press **D** to select the default WINDOW.

BROKEN WINDOWS

If you receive the message

PLEASE SPECIFY POINTS CORRECTLY!

then you've not specified the two corner points in their proper upper-left, lower-right order, or you've tried to set the WINDOW to an area on the Tablet that's too small. You will be asked to specify both corners again. If you want to cancel the attempted WINDOW, press **RETURN**.

If the Apple flashes the message

PLEASE STAY WITHIN THE WORK-AREA.

then you've let the pen stray outside the working area of the Tablet's overlay. You will be prompted again to indicate the corner point. To cancel the WINDOW command, press **RETURN**.

DRAWING IN THE WINDOW

Once you've placed a WINDOW on the Tablet, you can use any of the Tablet's drawing modes (DRAW, LINES, DOTS, FRAME, or BOX) to draw, as long as you stay within that WINDOW.

You can set the VIEWPORT within the WINDOW on the screen. Once you've set it, you can even REDUCE into it, and use the entire Tablet area within the VIEWPORT. When you turn the REDUCER off, you will again be limited to your WINDOW.

Once you've set a WINDOW, the only way to remove the WINDOW frame is to set a new WINDOW or use the Tablet RESET command. No other Tablet command will remove a WINDOW. The REDUCER will allow you to temporarily override the WINDOW; when you turn off the REDUCER, you will be left with the previous WINDOW again. Experiment with using WINDOW and the REDUCER; you'll be surprised at what they can do.



When you specify a WINDOW on the Tablet, the Apple will draw the WINDOW frame on the screen on top of the current picture. The sides of the frame are two dots wide, and the top and bottom are one dot wide. If you set a new WINDOW, the Apple will remove the frame by drawing over it with the BackGround COLOR. The WINDOW command can

therefore destroy parts of your previous picture. Also, even though WINDOW sets the VIEWPORT to the portion of the screen inside the WINDOW frame, the CLEAR command will clear the entire screen, including everything outside the VIEWPORT and the WINDOW frame, and even the frame itself! (The frame will be redrawn after the CLEAR.) If you reset the VIEWPORT to a slightly smaller size than the WINDOW, the CLEAR command will work normally and erase only what is within the VIEWPORT.

The WINDOW information is stored along with the picture information when you SAVE a picture onto diskette. There is no way to avoid saving this information. When you LOAD a picture, the Apple will automatically use the WINDOW setting stored with that picture, if that picture's diskette file name begins with PIC. If it does not, the Apple will use the default WINDOW setting for your Tablet. You can force the Apple to ignore the WINDOW setting stored in a picture file in three ways:

1) Leave the Graphics Tablet Software (see GETTING OUT) and RENAME the file, removing the PIC. prefix from the file name.

-- or --

2) When you LOAD the picture, type the PIC. flag at the beginning of the picture name. (LOAD PIC.filename)

-- or --

3) Once you have LOADED the picture, press the pen to WINDOW and type **D** to get the default setting.

RESET

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALI- BRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	-------------	-------	---------------	--------------	----------------	---------	--------------	------

The RESET command lets you "wipe the slate clean" and begin anew on a fresh picture. Namely, it:

- 1) Sets the WINDOW to the the normal 11 inch wide, 6.5 inch tall rectangle at the top of the working area.
- 2) Sets the VIEWPORT to the full screen.
- 3) Sets CALIBRATE to one unit per screen dot, and leaves the unit type undefined.
- 4) Sets the BackGround COLOR to black, and clears the screen.
- 5) Sets the PEN COLOR to white and sets DRAW mode.

- 6) Sets the value of DELTA to 2 and turns the Audio Feedback feature OFF.
- 7) Sets the default drive number for LOAD, SAVE, and CATALOG to 1.

Using the RESET command is just like restarting the Graphics Tablet software all over again.

A SOFTER RESET

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	-----------	---------	-----------	------

The SOFT RESET command is a milder version of RESET. It lets you reset many of your drawing and calculating functions, while leaving your picture, WINDOW settings, and pen color intact. SOFT RESET:

- 1) Sets the VIEWPORT to the full screen, or to the currently set WINDOW. This is the same as pressing **D** while setting a VIEWPORT.
- 2) Sets the CALIBRATE unit to 1 and the unit type undefined.
- 3) Sets the DELTA value to 2 and turns the Audio Feedback feature OFF.
- 4) Sets DRAW mode.

Nothing else is changed by SOFT RESET. The PEN COLOR, BackGround COLOR, the WINDOW setting, the default drive number, and so on, all remain the same.

CALIBRATE

RESET	CLEAR	WINDOW	BG COLOR	DELTA	SOFT RESET	VIEW PORT	CALIBRATE	REDUCER	PEN COLOR	DRAW
-------	-------	--------	----------	-------	------------	-----------	-----------	---------	-----------	------

The CALIBRATE command lets you specify a distance on the Tablet surface and use it for measuring with the DISTANCE and AREA functions.

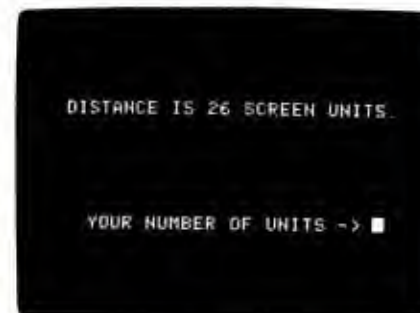
Press the pen to the CALIBRATE square on the menu. The Apple will beep and the brief question:

BEGINNING POINT?

will appear at the bottom of the screen. Select a point and press down. Another brief question:

ENDING POINT?

will flash at the bottom of the screen. Select another point, say, an inch away from the first and press down. The screen will vanish and the following frame will appear:



The Apple has converted the distance between the two points you specified into its internal "screen units". You now have the opportunity to define how long that distance actually was. If you've just arrived from Alpha Centauri, and you specified a distance of about one U.S. inch, then that's about 5 glibbets. Type

5 **RETURN**
G L I B B E T S **RETURN**

You've now defined the length you specified to be 5 Alpha Centauri glibbets. You're free to change it, of course, and give a distance of 10 chronacs, 200 malms, or even half a greton if you so desire (use decimal numbers for fractions, in this case **.5** gretons).

The CALIBRATE command won't let you specify negative distances, or distances greater than 999999999. It also won't let you use a name for your measurement which is more than 10 letters, numbers, or special characters (such as asterisks, bracketts, etc.) long.

You can tell the Apple that you'll agree to use its internal screen units for measurement by answering its questions about length and name with the **RETURN** key. Once you've specified a distance and name in the CALIBRATE command, they will remain as you set them until you: a) reset them with the CALIBRATE command, b) do a RESET or a SOFT RESET, or c) change the BackGround COLOR, the VIEWPORT setting, or the WINDOW setting.

MISTAKEN CALIBRATION

If you specify an endpoint for the CALIBRATE distance which is outside the current VIEWPORT, you will be asked to indicate the point again.

Once you begin to define a distance for the CALIBRATE command, you can cancel the procedure by pressing the **RETURN** key.



Don't change the REDUCER setting after you've CALIBRATED your Tablet. If you do, it will shrink your measurements just as it shrunk your Tablet movements, and all your DISTANCE and AREA calculations will be incorrect.

LONG DISTANCE...

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

Once you've set a distance and a unit name with the CALIBRATE command, you can use those definitions to calculate the DISTANCE that you move the pen along a path on the Tablet surface.

For example, find a road map (we'll use one of Central California), unfold it, and tape it to the Tablet so that the legend (with the scale of distance) is in the working area. Use the CALIBRATE command to set the distance and unit name to the scale of distance on the map:



Now point the pen to the DISTANCE square and press down. The Apple will beep, signaling you to take the pen and trace a path on the map. Draw a path from Buttonwillow to Bakersfield, along Route 5. The path

will appear on the screen as you draw. When you lift the pen, the Apple will beep again and flash

CALCULATING...

at the bottom of the screen, and then go away and think for a moment. It will soon return, telling you that the DISTANCE you traveled from Buttonwillow to Bakersfield is about 25 miles. After a short delay, you will be returned to DRAW mode.

The path you draw, as it appears on the screen, is just like any other path in DRAW mode, and it is subject to the same DELTA effects as DRAW. Lower DELTA settings will give you more accurate DISTANCES; higher DELTA settings will give you less accurate (albeit quicker) approximations.

If you invoke the DISTANCE command and then decide you don't want to calculate a distance after all, simply press the **RETURN** key instead of drawing a path on the Tablet.

There is a limitation on the maximum DISTANCE your path can be. The longest path you can draw for DISTANCE contains 800 points. With a DELTA setting of 2, this is 1600 screen units, or about 59 actual inches on the Tablet. Of course, this will be different if you're using the WINDOW or REDUCER functions.

... AND AREA CODES

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

The AREA command is a counterpart to the DISTANCE command. But instead of letting you find the DISTANCE between Buttonwillow and Bakersfield, CA, it will let you figure the AREA of Manhattan. Quickly remove the map of Central California and switch to one of the New York City area. Use the CALIBRATE command with the scale of distance on the new map.

Now place the pen on the AREA square and press down. The Apple will respond with a beep. Trace the perimeter of Manhattan. As soon as you lift the pen, the Apple will beep again and flash

CALCULATING...

and sit and think for a few moments. Soon it will return with the area of the island, expressed in the units you set in the CALIBRATE command. It will hold this value on the screen for about five seconds, and then return you to DRAW mode.

Now CLEAR the screen and try it again. This time, don't go completely around the island, but stop about half an inch away from your starting point. The Apple will obligingly close the curve for you, connecting the ending point directly to the beginning, before it calculates the AREA.

AREA is subject to the same limitations as DISTANCE: you can only draw a path 800 points long, or about 59 Tablet inches with a DELTA setting of 2. As in DISTANCE, a larger DELTA setting will give you less accurate results. And if you had the REDUCER on when you CALIBRATED, don't turn it off when you are going to calculate an AREA. A press of the **RETURN** key will abort the AREA command, just as it will for the DISTANCE command.

Now CLEAR the screen and find the AREA of Manhattan again. This time, go around the island twice. Notice that even though the AREA looks the same on the screen, the number that the Apple will return is about twice the actual AREA of the figure. This is normal: if you go around three times, the Apple will give you a number three times too large, and so on.

If, while drawing around an AREA, you move the pen outside the VIEWPORT, the Apple will act as if you had lifted the pen at that point, close the curve, and figure the AREA.

SLIDE RULES

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

Once you've got a picture on the screen, you don't have to rest at that. No, you can mobilize your pictures, give them some motivation, see some action! Press the pen to the SLIDE square. The request

BEGINNING POINT?

will appear briefly at the bottom of the screen. Use the pen to select any point on your picture, and press down. A second request

ENDING POINT?

will appear. Select another point on the screen, some distance removed from the first. Watch your picture travel across the screen, both vertically and horizontally, until the first point you selected (on the picture) is in the vicinity of the second point (on the screen).

If you decide that a SLIDE isn't what you want right now, press **RETURN** to cancel the operation. You'll be returned to your picture, in DRAW mode.

The SLIDE operation is performed in four directions, with what mathematicians call "toroidal wrap-around". This ponderous phrase means that the picture thinks it's not on a flat screen, but wrapped around a doughnut: the left side is joined to the right side, and the top is joined to the bottom, so that everything that you SLIDE off one edge of the screen will reappear on the opposite edge. When your SLIDE is complete, you will be returned to DRAW mode.

SLIDE moves the entire screen: there is no way to move only a portion of the screen. Because of the way the Apple places colors on the screen (see A SHORT DIGRESSION...), the SLIDE command can move the picture the exact vertical distance you indicate, but can only come within 14 dots of the horizontal location you specify.

SLIDE will remove the VIEWPORT and WINDOW borders before it moves the picture, but will replace them in their former locations (not SLID over) after the SLIDE is complete.

PRISMATIC APPLE

LINES	DOTS	FRAME	BOX	CATALOG	LOAD	SAVE	SEPARATE	SLIDE	AREA	DISTANCE
-------	------	-------	-----	---------	------	------	----------	-------	------	----------

The SEPARATE function "strips" your picture, until only one color is left. Press the pen to the SEPARATE square. You will be presented with a color menu, just like in PEN or Background COLOR. REMEMBER: The SEPARATE command will destroy parts of your picture. If you want to preserve a picture, be sure to SAVE it before you do a SEPARATE. If you've already started a SEPARATE command, just press **RETURN** to cancel it and return you to DRAW mode.

If you do want to SEPARATE out your picture, select a color from the color menu with the pen and press down. The menu will vanish and your picture will reappear. Quicker than you can pronounce "refraction", your picture will be stripped of all colors except the one you selected. You will be left in DRAW mode, with your Background COLOR set to black and your PEN COLOR set to the SEPARATE color you specified.

There is no way to undo a SEPARATION. The SEPARATE command will remove any VIEWPORT or WINDOW before it performs its function, and restore them when it's finished. SEPARATE works only on the entire screen: there is no way to SEPARATE only a portion of the screen.

You cannot SEPARATE out the color black. If you did, you'd be left with a blank screen! The Apple will deny your attempt to separate out either of the blacks with the message

NO SEPARATION ON BLACKS.

IN CONCLUSION

Congratulations! If you've come this far, and practiced with your Tablet along the way, then you've mastered the basic functions of the Apple Graphics Tablet. With a little practice, you can be drawing and manipulating pictures with skill and ease. If you're interested in doing more with your Tablet, and you're accustomed to programming in Applesoft BASIC, then you might be interested in looking into Chapter 3. It describes the internal operation of the Graphics Tablet software, and the operating subroutines in the Graphics Tablet itself. You'll find dozens of new applications for your Tablet. Go ahead, keep drawing, and have fun!

CHAPTER 3 PROGRAMMING THE GRAPHICS TABLET

46	The Programs
47	Tablet-Code Applesoft
49	The Main Loop
50	Menu Items
51	Extending the Menu
51	Example: Instant Color Menu
53	Example: Circle Mode
55	Example: Disc Mode
56	The Firmware
57	Tablet Control
60	Accepting Input
61	From Machine Language
64	Quick-Draw
65	By Any Other Name

THE PROGRAMS

There are four main programs which comprise the Graphics Tablet software. Three of these programs are supplied on your GRAPHICS TABLET SOFTWARE diskette, and the fourth is stored in ROM (Read-Only Memory) on the Tablet Interface card. These programs are:

- 1) **TABLET-CODE APPLESOFT:** This is a large applications program, written in the Applesoft II BASIC programming language. This is the program which performs all the commands and functions of the Tablet as described in Chapter 2.
- 2) **QUICK-DRAW:** This is a machine language subroutine which is used by the TABLET-CODE APPLESOFT program. This subroutine allows an Applesoft program to draw lines on the Apple's high-resolution graphics screen as fast as the Tablet can supply the points. This machine-language subroutine is hidden inside an Applesoft program.
- 3) **Tablet Firmware:** This is a set of subroutines permanently stored in ROM on the Tablet's Interface card. These are base-level subroutines for the basic operation of the Tablet. They can be used from any Apple programming language.
- 4) **UTILITIES:** This is a package of machine-language subroutines which perform many of the screen manipulation functions of the TABLET-CODE program. This package includes the subroutines which perform the SEPARATE and SLIDE operations. It also includes the shape table used by the Applesoft DRAW command to draw the corner marks for the VIEWPORT. These subroutines are stored in a binary file on the diskette and load at location \$6000 (decimal 24576) in memory. The length of this file is \$330 (816 decimal) bytes.

In addition, there are two other Applesoft programs which are used as part of the Graphics Tablet software package, but don't take part in the actual operation of the Tablet. They are:

- 1) **HELLO:** This is the program which is executed when you boot the diskette. It is also executed whenever you exit the TABLET-CODE or MENU ALIGNMENT programs. It allows you to select which program you wish to run, and lets you quit if you want to.
- 2) **MENU ALIGNMENT:** This is another Applesoft program that sets up an information file on the diskette, called TAB.INFORMATION. This file contains information about what slot the Interface card is in and where the overlay is located on the Tablet.

TABLET-CODE APPLESOFT

This is the main operating program for the Graphics Tablet. It is written in Applesoft, and takes up 12K bytes of the Apple's memory. It resides between locations \$1000 and \$3FFF (decimal 4096 and 16383) of memory. It requires that your Apple have the Applesoft language in ROM or on a Language System Language card. It will not run with cassette or diskette versions of Applesoft.

A source listing of this program appears in Appendix D, along with an atlas of subroutines, variables, and special locations. Here is a brief map to the program:

<u>Lines</u>	<u>Function</u>
10-160	Initialization. This section reads the Tablet information file, sets up all pertinent Tablet parameters, and initializes and clears Page 2 of the Apple's high-resolution graphics screen. It also places the program and its variables in the proper locations in the Apple's memory, and loads the UTILITIES subroutines.
170-180	This is the main DRAW mode loop. These two lines take input from the Tablet pen and send it to the QUICK-DRAW subroutine to draw on the screen. The only way out of this loop is to press a key or press the pen outside the Tablet's working area (i.e., on the menu).
190-194	These lines are executed when you press a key during DRAW mode. If you press any key other than ESC , nothing happens. If you press ESC , you will be asked whether you wish to quit or not. Pressing any key other than Y will return you to DRAW mode. Otherwise, the HELLO program will be run.
200-290	These lines sense when you press the pen to the menu. Line 280 is the main menu vector table.
300-310	The CLEAR command.
330-410	The LOAD command.
420-520	The SAVE command.
530-540	A subroutine to input the disk drive number during LOAD, SAVE, and CATALOG.
550-560	The SOFT RESET command.
570-610	The CATALOG command.

620-640 The BackGround COLOR command.

650 The PEN COLOR command.

660 This line lets you reenter LINES, DOTS, BOX, or FRAME mode after a menu selection.

670-870 This subroutine draws the color menu for BackGround COLOR, PEN COLOR, and SEPARATE, and lets you select a color with the pen.

880 Color box low-resolution draw.

890-1120 The WINDOW command.

1130-1140 This subroutine resets the Tablet WINDOW information after a color menu selection.

1150-1290 The VIEWPORT command.

1300 This subroutine causes a 1.15 second delay. It is used to pause while the Apple is displaying a message on the screen.

1310-1320 This subroutine waits until either a key is pressed on the keyboard or the pen is pressed down, and then returns.

1330-1340 This subroutine draws or undraws the four corner marks for a VIEWPORT.

1350-1360 This subroutine draws a single VIEWPORT corner mark.

1380-1390 The REDUCER command.

1400-1440 Turns on the REDUCER.

1460-1560 The DELTA command.

1580-1660 LINES mode.

1680-1720 DOTS mode.

1740-1820 FRAME mode.

1840-1930 BOX mode.

1940-1950 This subroutine is called whenever you specify a point outside of the VIEWPORT for any of the four modes mentioned above.

1970-1980 The AREA command. This section is the drawing loop.

1990-2030 The calculation section for AREA.

2070-2080 The DISTANCE command. This section is the drawing loop.

2090-2120 The calculation section for DISTANCE.

2160-2290 The CALIBRATE command.

2300 A subroutine to blank out the four lines at the bottom of Page 2 of Text mode.

2310 A subroutine to display "BEGINNING POINT?" at the bottom of Page 2 of Text mode and wait for the pen to be pressed down.

2320 A subroutine to display "ENDING POINT?" at the bottom of Page 2 of Text mode and wait for the pen to be pressed down.

2330-2480 The SLIDE command.

2490-2580 The SEPARATE command.

2590-2600 This subroutine resets the Tablet firmware.

2610-2730 Error handling subroutines.

THE MAIN LOOP

The main programming loop of the TABLET-CODE program occurs in lines 170 through 290. Lines 170 and 180 are the main loop for the DRAW mode. The DRAWing is done by the CALL EP% in line 170. This activates the QUICK-DRAW subroutine, which reads the Tablet and draws on the screen. As it draws on the screen, it also places the coordinates for each point plotted into the two arrays called X% and Y%. It uses the variable N% as an index into these arrays, and uses the value of the variable D% as its DELTA value (see the section on the QUICK-DRAW subroutine). The QUICK-DRAW subroutine returns to the Applesoft program when one of four events occur:

- 1) A key on the keyboard was pressed.
- 2) The pen was pressed down outside the working area of the Tablet.
- 3) The pen was lifted after being pressed down.
- 4) One of the arrays X% or Y% was filled up.

When QUICK-DRAW terminates, the termination condition code (a number from 0 to 3) is stored in location 700, and the Applesoft program

places this value into the variable CD. Depending upon the termination condition, the program either reinitializes the DRAW mode, or branches to line 190 (to handle a keypress) or line 200 (to get a menu selection).

If the pen was pressed outside the working area, then lines 220 through 240 sense the pen's position again in relation to the menu. Two numbers are returned: X holds a number from 0 to 21 which corresponds to one of 22 horizontal positions across the menu, and Y holds a number (0 or 1) which corresponds to one of two menu rows.

The ON Y+1 GOTO 280,290 statement in line 250 selects between the top and bottom menu rows. The ON X+1 GOTO 140... in line 280 selects among the 22 menu items in the top row.

MENU ITEMS

Each menu item corresponds to a block of code (not a subroutine) in the program. After the code for each item performs its function, it executes either a GOTO 170 (to reinstate DRAW mode) or a GOTO 660 (to reenter the current mode). A list of the variables which are used in the program and a description of their function appears in Appendix B. Subroutines which are called by parts of the program also appear in that Appendix.

The four other modes (LINES, DOTS, BOX, and FRAME) are actually independent menu items which operate differently from DRAW mode. For an example, look at the code for the FRAME mode, in lines 1740 through 1820. Line 1740 resets the Tablet to accept points from the current WINDOW area, with a sparkling crosshairs cursor displayed. It also sets the variable CM (for Current Mode) to 3, for FRAME mode. Line 1750 loops until the pen is pressed onto the Tablet surface, and returns the coordinates of the point in X and Y. These coordinates can be used directly to plot onto the high-resolution screen.

Line 1760 checks to see if the point is inside the current VIEWPORT. If it is not, the subroutine at line 1940 is called. This subroutine checks that the point is in the menu area. If it is, it sets the variable RT to 1; if not, it sets RT to 0 and displays the message

POINT OUTSIDE VIEWPORT. RESPECIFY.

Back in line 1760, if a menu item was selected, the code causes a jump to line 220 (menu selection). If the point was outside the viewport, the mode is restarted by a GOTO 1750.

If the point was inside the VIEWPORT, then line 1780 plots the point on the screen and saves its coordinates in the variables TX and TY. Lines 1790 through 1810 go through the whole get-a-point procedure again, and get another point in X and Y. Finally, line 1820 actually draws the FRAME, and jumps back to line 1750 to get another two points.

EXTENDING THE MENU

If you can write programs for the Apple, then you can tailor the TABLET-CODE APPLESOFT program to your own liking. You can add extra functions and remove or modify existing functions. You can define your own menu selections, or you can even start from scratch and write your own programs to use the Graphics Tablet to do just about anything.



The TABLET-CODE APPLESOFT program uses almost all of the memory space allotted for it. If you wish to add a function to the code, you must delete some of the program to make room for it. If the program grows any larger, it will not work.

EXAMPLE: INSTANT COLOR MENU

If you're tired of having to wait for the Apple to redraw the color menu when all you want to do is change the PEN COLOR from white to black, here's a modification you can make to get instant changes in PEN COLOR. To do it, you'll have to sacrifice one of the Tablet's other functions. Since this will mean changing your TABLET-CODE program, it's important that you not work on the original backup diskette.

Type

NEW

LOAD TABLET-CODE APPLESOFT

to load the unmodified program. To make room for your new code, delete a function you don't use much (some good candidates for oblivion are SLIDE, SEPARATE, AREA, DISTANCE, and CALIBRATE -- they are special-purpose functions and their removal won't affect the rest of the program). Let's delete the SLIDE function. Type

DEL 2330,2480

Now you've got about 250 more bytes to use for your program. To replace the SLIDE command with a null command, enter the line

2330 GOSUB 1130: GOTO 170

to reset the Tablet and return to DRAW mode if you try to select the now-defunct SLIDE function.

What we'll do is let you choose a new PEN COLOR simply by poking one of the first eight squares in the second row of the menu. Label these squares on your menu with a pencil or pen:



Now for the programming part. In lines 250 through 280, the variables X and Y hold the coordinates of the menu box which was just selected. Y is set to 0 for the top row and 1 for the bottom, and X holds a number between 0 and 21 corresponding to the 22 boxes in each row. So, if you poke the pen to one of the eight boxes of the new color menu, Y will be set to 1 and X will be a number from 0 to 7, depending upon which of the 8 squares you poked. It just so happens that the eight colors in Applesoft's high-resolution graphics mode are numbered 0 through 7, and they are in the exact same order as the color names you wrote on the second row of the menu! Isn't that lucky? But, first we've got to handle Y. In line 250, if Y is equal to 1, the program goes to line 290, the null function. Let's replace that line with

```
290 IF X<0 OR X>7 THEN GOSUB 1130: GOTO 170
```

Now the null function is executed only if the pen was pressed in the second row and not in the first eight boxes. If the pen was pressed in one of the first eight boxes, the next line will be executed. So let's make the next line:

```
295 PC=X: HCOLOR=PC: GOSUB 1130: GOTO 660
```

This line sets the Pen Color to the value of X (remember, 0 through 7?) and sets the high-resolution COLOR to that value. Then it resets the Tablet and goes to line 660, which reenters the current mode.

And that's it! Now before you RUN it save this version onto a copy of the GRAPHICS TABLET SOFTWARE diskette. You might want to add a REM statement at the beginning, describing the change and date. When you SAVE the program, you must save it under the name TABLET-CODE APPLESOFT or you won't be able to use it. Since the version of TABLET-CODE APPLESOFT that is on your diskette is LOCKed, you will have to UNLOCK it before you can save the new version.

To use your newly modified program, type

```
RUN HELLO
```

press **ESC**, select **G**, and press **RETURN**. When the program is running, you can instantly change colors in midstream, from any mode, by pressing the pen to the box for that color.

EXAMPLE: CIRCLE MODE

Here's another change which is a little more extensive than the previous one. Here are two extra modes, counterparts to BOX and FRAME, which draw open and filled circles rather than rectangles. You'll specify the location and size of a circle by poking two points on the Tablet: the first one will be the center of the circle and the second will be on the perimeter. Since this will require some extra room, (if you haven't done it already) delete a function such as SLIDE. In fact, you can delete both SLIDE and SEPARATE and use their menu squares to set CIRCLE and DISC mode. On a copy (not either of your original copies) of the GRAPHICS TABLET SOFTWARE diskette, type

```
LOAD TABLET-CODE APPLESOFT
DEL 2330,2580
```

Now they're gone. Since CIRCLE and DISC need two points, just like BOX and FRAME, we can follow the example of those modes. Type

```
2330 REM ** CIRCLE MODE **
2340 GOSUB 1130: PRINT D$;"IN#";SL: CM=5
2350 RT=2: INPUT X,Y,Z: IF Z<>2 THEN POKE -16368,0: GOTO 2350
2360 IF X<X3 OR X>X4 OR Y<Y3 OR Y>Y4 THEN GOSUB 1940: IF
    RT=1 THEN 220
2370 IF RT=0 THEN 2350
2380 HPLLOT X,Y: TX=X: TY=Y
2390 RT=2: INPUT X,Y,Z: IF Z<>2 THEN POKE -16368,0: GOTO 2390
2400 IF X<X3 OR X>X4 OR Y<Y3 OR Y>Y4 THEN GOSUB 1940: IF RT=1 THEN 220
2410 IF RT=0 THEN 2390
```

At this point, the coordinates of the center of the circle are in TX and TY, and the coordinates of a point on the perimeter are in X and Y. Let's find the radius of the circle now:

```
2420 R=SQR((X-TX)^2+(Y-TY)^2)
```


With a little trigonometry, we know that the horizontal and vertical distance from the center of a circle to any point on the perimeter is given by the simple formulae

2440 DX=R*SIN(TH): DY=R*COS(TH): X=TX: Y=TY

where R is the radius (derived in line 2420) and TH is an angle from 0 to 2π . Furthermore, we know that this formula gives us not one, but eight points on the circle:

X+DX,Y+DY	X+DX,Y-DY	X-DX,Y+DY	X-DX,Y-DY
X+DY,Y+DX	X+DY,Y-DX	X-DY,Y+DX	X-DY,Y-DX

as TH ranges from 0 to $\pi/4$ where X,Y is the center of the circle. So, let's add a loop and the lines to plot the points on the perimeter.

```
2430 FOR TH=0 TO .7854 STEP 1/R
2450 HPLLOT X+DX,Y+DY: HPLLOT X+DX,Y-DY: HPLLOT X-DX,Y+DY:
      HPLLOT X-DX,Y-DY
2460 HPLLOT X+DY,Y+DX: HPLLOT X+DY,Y-DX: HPLLOT X-DY,Y+DX:
      HPLLOT X-DY,Y-DX
2470 NEXT TH: GOTO 2350
```

Now to finish it all up, change the lines

```
660 ON CM+1 GOTO 170,1580,1680,1740,1840,2330
1380 RD=RD+1: IF RD>1 THEN RD=0: GOSUB 1130: GOTO 660
1390 GOSUB 1400: GOTO 660
```

These changes let you go back to CIRCLE mode automatically after making a menu selection.

Now UNLOCK the old version of TABLET-CODE APPLESOFT that is on your diskette, (the one on which you're putting your own versions) and then SAVE this new version on your diskette. RUN HELLO and start using the GRAPHICS TABLET SOFTWARE. When you want to draw an open circle, press the pen to the square marked SLIDE. Indicate one point for the center of the circle, and another for a point on the perimeter. The circle will be drawn to specification, and you'll remain in CIRCLE mode until you choose another.

Be forewarned that if you make a CIRCLE which is too large for the screen, then you'll get an error. Just press **ESC** to get back to DRAW mode. If you don't like this "feature", the following lines will fix the problem:

```
2425 ON ERR GOTO 2480
2470 NEXT TH: ON ERR GOTO 2650
2475 GOTO 2340
2480 PRINT D$;"PR#0": GOSUB 2300: PRINT D$;"PR#";SL: PRINT "M2":
      VTAB 23: HTAB 12: POKE 41, PEEK(41)+4: PRINT "CIRCLE OFF SCREEN.
      RESPECIFY."
2485 GOSUB 1300: PRINT D$;"PR#";SL: PRINT "N,H2": ONERR GOTO 2650
2490 GOTO 2340
```

Note that you can still make circles which go out of the VIEWPORT. There's no easy way to prevent this.

EXAMPLE: DISC MODE

The DISC mode is just the same as CIRCLE mode, except that instead of plotting individual points on the perimeter, you'll have to draw lines across the diameter to fill in the circle. Because they have so much code in common, you can make DISC use much of the code from CIRCLE. Here are the changes to CIRCLE to make it do DISCs, too:

```
2340 CM=5: GOTO 2348
2342 REM ** DISC MODE **
2344 CM=6
2348 GOSUB 1130: PRINT D$;"IN#";SL

2445 IF CM=6 THEN 2464
2462 GOTO 2470
2464 HPLLOT X+DX,Y+DY TO X-DX,Y-DY: HPLLOT X+DX,Y-DY TO X-DX,Y+DY
2466 HPLLOT X+DY,Y+DX TO X-DY,Y-DX: HPLLOT X+DY,Y-DX TO X-DY,Y+DX
```

To change the menu vector table so that the SLIDE square will activate CIRCLE and the SEPARATE square will activate DISC, change line 280 to read:

```
280 ON X+1 GOTO 140,300,890,620,1460,550,1150,2160,1380,650,290,1580,
      1680,1740,1840,570,330,420,2342,2330,1970,2070
```

and make this other change:

```
660 ON CM+1 GOTO 170,1580,1680,1740,1840,2330,2342
```

This next line lets you reenter both CIRCLE and DISC modes after you make a menu selection (such as PEN COLOR). Finally, if you added the error handling subroutine described above, then change it so:

```
2390 ON CM-4 GOTO 2330,2342
```

Now again SAVE your modified program under the name TABLET-CODE APPLESOFT on the diskette you're using for your experimentation.

THE FIRMWARE

On the Graphics Tablet Interface card is a 2K byte ROM (Read-Only Memory). This ROM contains all the subroutines which read and interpret the signals from the Graphics Tablet. These subroutines can be used easily from any BASIC program.

The Graphics Tablet Firmware performs many functions. Its main purpose is to read the position of the Tablet's pen on the surface of the Tablet, and return that position in a numerical form to a BASIC program. But, it also does much more:

- It lets you supply horizontal and vertical offset information. It will use this offset information in calculating the pen position. This lets you place the origin (where the X and Y coordinates are both 0) anywhere on the Tablet surface. The offsets can be integers from -32767 to +32767.
- It allows you to give a scaling divisor, from 1 to 32767. You can tell the Tablet firmware to divide all coordinates by this number before it passes them to your BASIC program. This lets you calibrate the Tablet units (200 to the inch) to your own scale.
- It allows you to select among ten different modes on the Apple's screen. Text, low-, and high-resolution graphics (on either Page 1 or Page 2) can be selected, and you can mix text with graphics.
- It automatically displays a flashing cursor on the Apple's screen, given the proper scaling and offset information. Cursors are available or may be suppressed in all screen modes.
- You can tell the Tablet to suppress all output from your Apple.
- You can read not only the position of the pen, but also whether it is within readable distance, whether the pen is up or pressed down, detect pen-down and pen-up movements, and read the keyboard to see if a key has been pressed.

Your programs can communicate to the Firmware subroutines by using the BASIC commands `PR# s` and `IN# s`, where s is the number of the peripheral connector slot in the Apple which holds the Tablet Interface card. (The `PR#` command indicates that all subsequent output is to be directed to the Firmware subroutines in a certain slot, and the `IN#` command indicates that all subsequent input is to be taken from the Firmware subroutines in the given slot.) When your program wants to stop talking to the Firmware subroutines, it can issue a `PR#0` or `IN#0` command to direct output or accept input from the normal screen and keyboard.

To avoid alienating DOS (the Disk Operating System), you'll have to issue the `PR#` and `IN#` commands in the form of DOS commands. See the section on Selecting I/O Devices in your DOS manual.

TABLET CONTROL

To send control information to the Tablet, just execute a `PR# s` command from BASIC and PRINT a string of Tablet Control commands. The Control commands will not be displayed on the Apple's screen: they will be used by the Tablet alone.

There are seventeen Tablet Control commands. These commands take the form of a letter or a word, sometimes followed by a number. Commands are executed in a sequential order as given to the tablet by the user.

Commas are used as delimiters between commands and must not begin or end the command string. Spaces are ignored. A null string issued to the tablet is invalid. Only the first alphabetic character of a command is meaningful; the other alphabetic characters are ignored and may be omitted.

Following is a list of Tablet Control commands. The letter "n" that follows some of these commands represents an integer. The Tablet Control commands are:

TEXT n	Sets the Apple's screen to show text mode. n determines which page of Text to display and can be either 1 or 2.
HGR n	Sets the Apple's screen to show full-screen high-resolution graphics mode. n determines which page of graphics to display and can be either 1 or 2.
LGR n	Sets the Apple's screen to show full-screen low-resolution graphics mode. n determines which page of Graphics to display and can be either 1 or 2.
MIXHGR n	Sets the Apple's screen to show high-resolution graphics mode, mixed with four lines of text at the bottom. n determines which page of text and graphics to display and can be either 1 or 2.
GR n	Sets the Apple's screen to show low-resolution graphics mode, mixed with four lines of text at the bottom. n determines which page of text and graphics to display and can be either 1 or 2.

SCALE n Sets the Tablet scaling divisor to n. All coordinates generated by the Graphics Tablet will be divided by n before they are given to your program. The range for n is 1 to 32767. If you give the Tablet a negative scaling divisor, it will ignore the minus sign and use the positive number. A scale factor of 0 is undefined and will not work.

XOFF n Sets the Tablet horizontal (X) offset to n. If the R command is enabled, all horizontal coordinates will have n added to them before they are given to your BASIC program. The offset value, n, may range from -32767 to +32767.

YOFF n Sets the Tablet vertical (Y) offset to n. If the R command is enabled, all vertical coordinates will have n added to them before they are given to your BASIC program (see R command below). The offset value, n, may range from -32767 to +32767.

F Ignore scaling divisor. None of the coordinates generated by the Tablet will be scaled or offset. The cursor, however, will not ignore scale and offset information.

R Use scaling divisor. All coordinates generated by the Tablet will be divided by the scaling divisor before they are given to your BASIC program. Then offset values will be added.

AFTER If the R command is used, the offsets will be added after the scaling operation. This command is turned off (the BEFORE command is reinstated) with any subsequent command which sets a screen mode, including the DEFAULT command.

BEFORE If the R command is used, the offsets will be added before the scaling operation.

NOPRINT Disables all on-screen printing. After a NOPRINT command is sent to the Tablet, no new output generated by the Apple will be displayed on the screen. NOPRINT mode is turned off by any other Tablet Control command string or by a BASIC PR#0 command.

CURSOROFF Turns off the sparkling cursor. The cursor will remain off until any other Tablet Control command is sent which sets a screen mode (the DEFAULT command also turns the cursor on).

Q Sets Stream mode. If the pen is within the proximity of the Tablet, the Tablet Firmware will send coordinates each time it is polled, regardless of the pen position or status. This command is turned off (the Q command

is reinstated) with any subsequent command which sets a screen mode, including the DEFAULT command.

Q Resets Stream mode. The Tablet firmware will send coordinates only when it is polled, and the pen is pressed down.

DEFAULT Sets the standard (default) Tablet mode:

- HGR 2 screen mode
- SCALE=16
- XOFF=1536
- YOFF=1536
- F (no scaling or offsets)
- BEFORE
- Q (stream mode off)
- Cursor on
- Printing on

For example, if the Tablet Interface card is in slot number 5 and you want the Tablet to set low-resolution graphics mode, with four lines of Text at the bottom, use a scaling divisor of 16, and use the offsets stored in the variables X0 and Y0, and apply them before the scaling, you would use this Tablet Control command:

```
PR#5: PRINT "GR 1, SCALE=16, XOFF=";X0;"", YOFF=";Y0;"", BEFORE":PR#0
```

Of course, you could shorten it by eliminating extraneous spaces and using only the first letter of each Control command name:

```
PR#5: PRINT "G1,S16,X";X0;"",Y";Y0;"",B":PR#0
```

Since you are using Apple DOS, you must use DOS's PR# command in order to use both DOS and the Tablet. If you've got the slot number of the interface card in the variable S1, then the same Tablet Control command would read:

```
PRINT D$;"PR#";S1:PRINT "G1,S16,X";X0;"",Y";Y0;"",B":PRINT D$;"PR#0"
```

It's important not to add a semicolon (;) or comma (,) at the end of the Tablet Control PRINT string. The Tablet will execute the command only when it receives a RETURN character. A semicolon or a comma after the string will suppress the RETURN; therefore, the Tablet will never carry out your commands because it won't hear the end of them.

Any illegal construct in a Control command, including numbers out of range, will cause the screen to return to text mode and the message

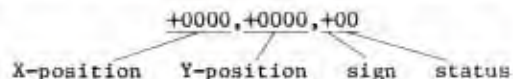
*** TABLET SYNTAX ERROR

to appear on the screen.

ACCEPTING INPUT

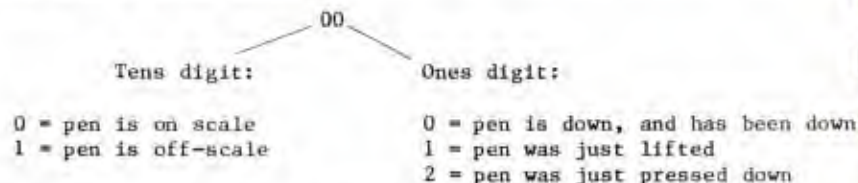
Once you've told the Tablet what kinds of numbers you expect to be getting from it, you can use the BASIC statements IN# and INPUT to get the pen coordinates and status information from the Tablet.

The Tablet sends its coordinate and status information in this format:



The X- and Y-position coordinates must be integers from -9999 to +9999. The user is responsible for adjusting the X and Y offsets and the scale value so that values returned by the tablet fall within this range. It is possible to exceed this range, in which case an Applesoft error will be generated. These coordinates indicate the position of the pen on the Tablet. If the R command is in effect, these coordinates indicate the position of the pen plus the offset and divided by the scaler.

The sign and status digits indicate the status of the pen and keyboard. If the sign is negative, then a key has been pressed. The two digits have separate meanings:



So let's write a program to read the Tablet and print out the coordinates, without scaling or offsets, on the Text screen. Let's assume that the slot number of the Interface card is stored in the variable SL.

```
100 PR#SL: PRINT "T1, F, C, P": PR#0: REM INITIALIZE TABLET
110 IN#SL: INPUT X,Y,Z: IN#0: REM READ TABLET
120 PRINT "THE X-POSITION IS ";X;"", "THE Y-POSITION IS ";Y;"."
130 IF Z<0 THEN PRINT "THE KEYBOARD HAS BEEN PRESSED."
140 IF ABS(Z)>=10 THEN PRINT "THE PEN IS OFF-SCALE."
150 Z=ABS(Z): IF Z>=10 THEN Z=Z-10: REM GET ONES DIGIT
160 IF Z=0 THEN PRINT "PEN IS DOWN."
170 IF Z=1 THEN PRINT "PEN WAS JUST LIFTED."
180 IF Z=2 THEN PRINT "PEN WAS JUST PRESSED DOWN."
190 PRINT
200 POKE -16368,0: GOTO 110:REM CLEAR KEYBOARD STROBE, REPEAT
```

This program will work in either Applesoft BASIC or Apple Integer BASIC.

Line 100 sets the Tablet Control parameters. Line 110 gets input from the tablet, and the remaining lines interpret the values and print an explanation. Line 200 clears the keyboard strobe (if a key was pressed) and loops back to get another set of values.

This program works in Stream mode, that is, it is continually getting input from the Tablet regardless of the position of the pen. If you change the Tablet Control command string to read

```
100 PR#SL: PRINT "T1, F, C, Q": PR#0
```

then the coordinates will be returned only when the pen is pressed down.

Let's write a subroutine in BASIC which is to return the X and Y coordinates of the next pen press, or return with the variable KY set to 1 if the user presses the **RETURN** key on the keyboard. Let's assume that the Tablet has been initialized in the main program (see previous example, line 100).

```
200 REM ** SUBROUTINE TO GET A PEN PRESS OR KEYPRESS **
210 KY=0: REM FLAG FOR KEYPRESS
220 IN#SL: INPUT X,Y,Z: IN#0
230 IF Z=2 THEN RETURN: REM PEN DOWN
240 IF Z>0 THEN 280: REM NO KEYPRESS
250 K=PEEK(-16384): REM GET KEYPRESS
260 IF K<>13 THEN 280: REM IS IT A RETURN?
270 KY=1: RETURN: REM YES, IT IS.
280 POKE -16368, 0: GOTO 220: REM NO, KEEP LOOKING.
```

FROM MACHINE LANGUAGE

You can perform the same Tablet operations from within a machine language program that you can from a BASIC program. Even though machine language programs are a little more difficult to write, they will run faster and use less memory than their BASIC counterparts.

Your machine language programs will invoke the various functions of the Graphics Tablet firmware by performing JSR (Jump to SubRoutine) operations to subroutines inside the Tablet's ROM, rather than using the PR# and IN# statements in BASIC. Your programs will pass information to the Tablet by storing it in fixed locations in memory, and will receive information from the Tablet by storing it in other fixed locations, instead of using PRINT and INPUT statements as a BASIC program would.

Since the Tablet firmware operates in the same manner regardless of whether it is being driven by a BASIC or a machine language program, this section will explain only the specifics of machine language operation of the Tablet. For a description of the modes and parameters which the Tablet firmware recognizes, please see the previous section.

The Tablet firmware is absolutely located in the Apple's memory at locations \$C800 through \$CFFF. This is a 2K memory space which is shared by all peripherals, and can be used by any one peripheral card at any time. In order to let the Graphics Tablet card take possession of this common ROM space, you must reference two special memory locations. First, you must reference location \$CFFF. This will turn off all interface cards which may be using the common ROM space. Then you must make at least one reference to any address in the range \$Cn00 through \$CnFF, where n is the number (from 0 to 7) of the peripheral connector slot which holds the Graphics Tablet interface card. Once this is done, the Tablet's ROM will be placed into its proper memory range and you can reference its subroutines normally.

After you activate the ROM, you should store the slot number of the Graphics Tablet (in the format \$Cn) in location \$07F8. This lets other Apple programs know that the Tablet is active and in use.

Subroutine POINT (location \$Cn02) lets you read a single point from the Tablet. The coordinates of the point, along with the pen status information, will be stored as a 15-character long ASCII string, beginning at location \$0200 and ending with a RETURN code at location \$020E. The format of this string is described in the previous section called ACCEPTING INPUT.

The subroutine DEFAULT (location \$CE90) sets all the Graphics Tablet parameters and modes to their default values. It operates the same as the Tablet control command DEFAULT.

The subroutine MREAD (location \$CBB9) allows you to read the pen position and status quickly, and get the result in binary (rather than ASCII, as POINT does). It returns the X and Y coordinates in the following locations:

XFFL	\$0281	Lower byte of X-coordinate
XFFH	\$0282	Upper byte of X-coordinate
YFFL	\$0283	Lower byte of Y-coordinate
YFFH	\$0284	Upper byte of Y-coordinate
TEM	\$0280	Pen status

The X and Y coordinates are numbers from -32767 to +32767. Notice that this is a greater range than the coordinates passed by POINT. The numbers are in two's complement form, and the high bit of the upper byte of each coordinate determines the sign of that coordinate. The pen status byte is interpreted much the same as it is for POINT: the lower 4 bits represent the pen status and the upper bit represents the keyboard status.

The SCALE subroutine (location \$CB70) is normally called immediately after MREAD. It performs a scaling and offset operation on the X and Y coordinates generated by MREAD and places the results in these four locations:

TEMXL	\$0285	Lower byte of scaled X-coordinate
TEMXH	\$0286	Upper byte of scaled X-coordinate
TEMYL	\$0287	Lower byte of scaled Y-coordinate
TEMYH	\$0288	Upper byte of scaled Y-coordinate

These values are also in two's complement form and range from -32767 to +32767.

The CURSOROUT subroutine (location \$CBF0) is normally called immediately after an MREAD. CURSOROUT calls SCALE and uses the scaled results to place a cursor on the Apple's screen. The cursor is placed by an exclusive-OR operation, so another call to CURSOROUT using the same coordinates will remove the cursor and leave the screen unchanged.

The CURSOROUT subroutine places the cursor on the screen which the Tablet was told to display. It is not necessarily the screen which the Apple is currently displaying. If you manually change the screen setting after calling DEFAULT or setting the Tablet PAGE parameter (see below), then the Apple may be displaying a video mode which is different from the one in which the Tablet is displaying a cursor.

You can pass parameters to the Tablet firmware by storing the proper values in special memory locations. Here are the locations used by the Tablet firmware.

The MSL0T parameter (location \$07F8) contains the number of the slot (in the format \$Cn) into which the Graphics Tablet Interface card is plugged.

The PAGE parameter (location \$03B8+MSLOT) holds the code for the current video mode:

\$20	high-resolution page 1	\$40	high-resolution page 2
\$01	low-resolution page 1	\$02	low-resolution page 2
\$21	Mixed high-resolution page 1	\$42	Mixed high-resolution page 2
\$05	Mixed low-resolution page 1	\$0A	Mixed low-resolution page 2
\$04	Text page 1	\$08	Text page 2
\$00	No cursor		

If you set the high bit of the PAGE byte, then the scale and offset factors will be applied.

The MPAGE parameter (location \$0438+MSLOT) holds some of the same information as the PAGE parameter. The lower six bits of MPAGE are derived from the lower six bits of PAGE exclusive-ORed with the constant \$25. The upper two bits represent the A, B, P, and Q parameters:

Bit 7 ON: Stream mode on	Bit 7 OFF: Stream mode off
Bit 6 ON: Offset after scale	Bit 6 OFF: Offset before scale

The scale and offset parameters are stored in the following locations:

SCALL \$0488+MSLOT	Lower byte of scaling divisor
SCALB \$0538+MSLOT	Upper byte of scaling divisor
OFFXL \$05B8+MSLOT	Lower byte of X-offset
OFFXH \$0638+MSLOT	Upper byte of X-offset
OFFYL \$06B8+MSLOT	Lower byte of Y-offset
OFFYH \$0738+MSLOT	Upper byte of Y-offset

The scaling divisor is a binary integer from 0 to 32767. The offsets are two's complement binary numbers from -32767 to +32767.

QUICK DRAW

The QUICK-DRAW program is a machine language subroutine which acts as an intermediary between the Tablet Firmware and an Applesoft program. Since an Applesoft program using HPLLOT cannot draw on the High-Resolution screen fast enough to keep up with the movements of the pen across the Tablet, the QUICK-DRAW subroutine talks directly to the Tablet and plots the points on the high-resolution screen. QUICK-DRAW also makes the points plotted available to the Applesoft program.

QUICK-DRAW must run on an Apple with at least 16K bytes of memory, the Applesoft II BASIC programming language in ROM or the Language System, and a Graphics Tablet Interface card. The Graphics Tablet Firmware must be activated by an IN# command before QUICK-DRAW can be called.

The QUICK-DRAW subroutines are hidden inside an Applesoft program. When you RUN QUICK-DRAW, the Applesoft program will store the subroutines in the memory range \$C00-\$FFF (decimal 3072-4095). The entry point for the subroutines will be placed in memory locations \$2F0 and \$2F1 (decimal 752 and 753). Your Applesoft program, which you will RUN right after you RUN QUICK-DRAW, can PEEK at these locations and get the entry point by executing this line:

```
100 EP%=PEEK(752)+256*PEEK(753)
```

The QUICK-DRAW subroutine deals directly with four Applesoft variables. When you CALL the QUICK-DRAW subroutines, it takes the coordinates of the points it receives from the Tablet and places them in the two Applesoft arrays X% and Y%. It uses the Applesoft variable

N% as an index into these arrays. The subroutine also uses the contents of the variable D% as a DELTA value. It is the QUICK-DRAW subroutine which controls the DELTA and Audio Feedback features of the Tablet software.

You must dimension the arrays X% and Y% prior to calling QUICK-DRAW. Also, you must assign a non-zero value to D%. The D% value is used as described in the DELTA function in Chapter 2; if the value of D% is negative, then the Audio Feedback feature will be turned off.

The QUICK-DRAW subroutine will return control to the Applesoft program under any of four conditions:

- 1) A key on the keyboard was pressed before the pen was pressed down.
- 2) The pen was moved to a place on the Tablet which does not correspond to a position in the current VIEWPORT.
- 3) The pen was lifted after being pressed down inside the VIEWPORT.
- 4) There is no more room in the arrays X% and Y% to store coordinate values.

When one of these conditions arises, the code for that termination condition will be stored in location \$2BC (decimal 700) and control will be returned to the Applesoft program.

You can define a VIEWPORT for the QUICK-DRAW subroutines by storing:

- the coordinate of the left edge in locations 3089 and 3090;
- the coordinate of the right edge plus one in locations 3091 and 3092;
- the coordinate of the top edge in location 3093; and
- the coordinate of the bottom edge plus one in location 3094.

See lines 1100 and 1120 of the TABLET-CODE APPLESOFT program for an example of how to pass VIEWPORT coordinates to the QUICK-DRAW subroutine.

BY ANY OTHER NAME

You can change the names of the variables which QUICK-DRAW will use by executing a special CALL to QUICK-DRAW. Normally, QUICK-DRAW uses these variable names:

D% for the DELTA value	N% for the index to the arrays
X% for the X-coordinate array	Y% for the Y-coordinate array

You can change these four variable names to be whatever you like. However, they must always be of the integer variable type (denoted by the percent sign (%) following the name). To rename the variables, use this format:

```
220 CALL EP%,DELTA%,NUMBER%,XVAL%,YVAL%
```

Since Applesoft only recognizes the first two letters of a variable name, this will make QUICK-DRAW use the variable DE% for its DELTA, NU% for N%, XV% for X%, and YV% for Y%. If you want to change only one of the names, just leave the others out, but keep the proper number of commas:

```
230 CALL EP%,IN%,,
```

will make QUICK-DRAW use the variable IN% instead of N%. You must keep the variable names in the order D%,N%,X%,Y%.

APPENDIX A USE AND CARE

68	Care of the Menu Overlay
68	Care of the Tablet
69	Care of the Interface
69	If It Doesn't Work

CARE OF THE MENU OVERLAY

You can write on the clear plastic menu overlay with most anything: soft (Number 2) pencils, felt-tip pens, permanent markers, crayons, and the like. However, ball-point pens tend not to write well on the overlay, and colored or hard lead pencils also have problems.

You can wipe the overlay clean of most marks or doodles you have drawn using a soft cloth and a mild soap-and-water solution. Most marks from felt-tip or "permanent" markers can be removed easily. Some markers, however, will leave truly permanent scars on the overlay; it's a good idea to test any marker on an inconspicuous corner of the overlay before you use it to draw all over your Tablet. To be safe, use felt-tip markers designed for use on acetate or mylar (or for use with overhead projectors). These will give you visible, non-smearing colors, but the marks will wipe off without a trace.

If you are getting inexplicable "glitches" on your screen you probably have a static problem. The solution is simple: Wipe the overlay with the static cloth that came with your Graphics Tablet. A treatment with the cloth should remove any excess static from the overlay.

CARE OF THE TABLET

Your Graphics Tablet is constructed of a solid wood base, protected below by a sheet metal baseplate and above by a molded, snap-on plastic cover. If the top cover gets dirty, it can be cleaned with a soft cloth and a mild soap-and-water solution. Don't use any abrasives or strong detergents on the surface or case of the Tablet: they may scratch or damage the plastic. If possible, keep the Tablet covered when you aren't using it.

DON'T leave anything which has a strong magnetic field on or near the Tablet. This will disrupt its natural magnetic orientation and make it malfunction. Keep your diskettes off the Tablet! Its magnetic field may alter or erase the information on them. Don't place disk drives, televisions, electric motors, magnets, or large, heavy metallic objects on top of the Tablet.

Keep the Tablet in a cool, dry place. Don't leave it in direct sunlight or in a car trunk or some other hot, stuffy place. Too much heat will warp its cover.

Be careful with the Tablet when you're moving it from place to place. Don't drop it or jar it. Even though it's pretty solid, it can be seriously damaged by a bad fall.



CARE OF THE INTERFACE

The Interface card is really the most delicate part of the Graphics Tablet. When inserting, removing, or transporting it, be very careful not to bend any of its pins or components. To be safe, always carry it in the box in which it was shipped, nestled in protective foam. Keep it away from strong electrical or magnetic fields, and don't even think of touching it if there's a lot of static electricity in the area.

If you've been inserting and removing the Interface card into the Apple a lot, then it's possible that the metal "fingers" have gotten dirty and are not making good contact with the Apple. In this case, the easiest way to clean the fingers is to just use an ordinary pencil eraser and rub all of the gunk off. If you want to be thorough, use cotton swabs and rubbing alcohol to clean the fingers on the card.

IF IT DOESN'T WORK

If you've exposed your Tablet to bad magnetic influences or it's been bumped and jarred a lot, it may develop "dead spots" on its surface, spots where the pen won't draw. These aren't permanent, they're just a loss of magnetic orientation in certain spots of the Tablet. Take the Tablet to your Apple service center. The service center should have the proper equipment to reorient your errant Tablet and make it work again.



APPENDIX B BACKING UP YOUR DISKETTES

72 With Two Disk Drives

72 With One Disk Drive

WITH TWO DISK DRIVES

If your Apple has two disk drives, you can easily make a copy of either GRAPHICS TABLET SOFTWARE diskette by using the diskette copying program on your Apple SYSTEM MASTER diskette. You will need three diskettes:

- 1) One of the GRAPHICS TABLET SOFTWARE diskettes, enclosed with your Tablet;
- 2) The SYSTEM MASTER diskette, enclosed with your Disk II; and
- 3) A blank, uninitialized diskette. If you like, you can use a preinitialized diskette, but all information on that diskette will be destroyed.

Boot your system using the SYSTEM MASTER diskette (see your DOS manual, or, if you have an Autostart ROM, see your Autostart ROM manual) and type

RUN COPY

After the disk drive stops whirring, place the GRAPHICS TABLET SOFTWARE backup diskette in one drive, and place the blank diskette in the other. The GRAPHICS TABLET SOFTWARE diskette will be the "Original", and the blank diskette will be the "Duplicate". Follow the instructions in the section on using the COPY program in your DOS manual.

Once you've copied the diskette, label the duplicate so you'll know what it is. Then put the original away in a safe place. If you ever lose or destroy the duplicate, then before you start to use the original, make another copy of it. It's also a good idea to periodically make duplicate copies of the diskettes which hold your pictures.

WITH ONE DISK DRIVE

If your Apple has only one disk drive, then you'll have to copy all the programs which comprise the GRAPHICS TABLET SOFTWARE package one by one, loading each program from the original diskette and saving it to the duplicate. It's a lengthy procedure, but well worth your trouble.

Boot your system using the GRAPHICS TABLET SOFTWARE diskette. Press **ESC** to get to the HELLO menu, select **Q** to QUIT, and press **RETURN**. Now remove the diskette and write-protect it by sticking a write-protect tab (a thin but sturdy strip of tape will do) over the

square notch on the left side of the diskette. This is important! It will prevent you from accidentally destroying anything on the original diskette. Now insert a blank, uninitialized diskette in the drive. You can use a preinitialized diskette, but all information on it will be destroyed. Type

INIT HELLO

and press **RETURN**. You're now initializing the diskette with the HELLO program from the GRAPHICS TABLET SOFTWARE diskette. This takes about a minute.

Now switch to the GRAPHICS TABLET SOFTWARE diskette and type
LOAD MENU ALIGNMENT

Now switch to the duplicate diskette and type
SAVE MENU ALIGNMENT

Now switch to the GRAPHICS TABLET SOFTWARE diskette and type
LOAD TABLET-CODE APPLESOFT

Now switch to the duplicate diskette and type
SAVE TABLET-CODE APPLESOFT

Now switch to the GRAPHICS TABLET SOFTWARE diskette and type
LOAD QUICK-DRAW

Now switch to the duplicate diskette and type
SAVE QUICK-DRAW

Now switch to the GRAPHICS TABLET SOFTWARE diskette and type
BLOAD UTILITIES
BLOAD GRAPHICS TABLET LOGO

Now switch to the duplicate diskette and type
BSAVE UTILITIES, A\$6000, L\$330
BSAVE GRAPHICS TABLET LOGO, A\$2000, L\$2000

Now enter this program:

NEW

```
10 D$=CHR$(4)
20 PRINT D$;"OPEN GRAPHICS TABLET SOFTWARE"
30 PRINT D$;"WRITE GRAPHICS TABLET SOFTWARE"
40 PRINT "RUN QUICK DRAW"
50 PRINT "RUN TABLET-CODE APPLESOFT"
60 PRINT D$;"CLOSE GRAPHICS TABLET SOFTWARE"
70 END
```


RUN

This short program creates an EXEC file called GRAPHICS TABLET SOFTWARE, whose function is to set up the Apple to RUN the programs which make the Graphics Tablet work. You need to have this file on every duplicate diskette you make; if you're going to be making many duplicate copies, you might want to SAVE this short program so you don't have to retype it every time you need it.

To SAVE this program, type

SAVE FILEMAKER

Then, whenever you're making a duplicate, put in this diskette and type

LOAD FILEMAKER

put in the duplicate diskette, and type

RUN

That's all there is. Once you've copied the diskette, label the duplicate so you'll know what it is and put it away in a safe place. If you ever lose or destroy the original, then before you start to use the duplicate, make another copy of it.

APPENDIX C COLOR ANOMALIES

- 76 Unusual Color Effects...
- 77 ...And How to Get Them

UNUSUAL COLOR EFFECTS...

You may have already noticed that a few strange things happen when you try to use certain combinations of colors with the Graphics Tablet. Don't worry: these are normal, predictable phenomena which are caused not by the Tablet, but by the Apple itself.

The Graphics Tablet displays its pictures using the Apple's high-resolution graphics mode. In this mode, there are 53,760 individual dots on the screen, and six colors (black, white, orange, blue, green, and violet). The Apple should therefore need several hundred thousand individual "bits" of information to form a picture. But the Apple uses only 65,536 bits of information (organized into 8,192 eight-bit "bytes") to form the picture! The reason the Apple can display such complex pictures using so little memory is the same reason that sometimes the colors don't appear normal: not all colors can be used in all places on the screen, and each dot is limited in the number of colors it can be.

This specialization of function causes some combinations of colors to work differently than you might expect. There are three different effects which are caused by the limitation in the color scheme. They occur in all drawing modes, but only where one color borders another and the borderline is not horizontal. For example, the color problems could occur on the two vertical sides of a FRAME but not on the top or bottom. Here are the three effects:

- 1) **DASHED LINES.** When you draw black or white lines on a colored field (or vice versa), non-horizontal lines will tend to become dashed and incomplete, and vertical lines may not appear at all.
- 2) **ZEBRA STRIPES.** When you draw colored lines on a colored field, non-horizontal lines don't appear their normal colors, but instead are sometimes black-and-white striped. Vertical lines will appear either completely black or completely white.
- 3) **COLOR FLIP.** When you draw with one color (or black or white) across a colored field, sometimes a seven-dot wide area around a non-horizontal line will change color. This will result in a colored "shadow" appearing around the line.

These effects occur in various combinations, depending upon the colors you use.

... AND HOW TO GET THEM

The table on the next page illustrates seven different combinations of the effects mentioned above, and what color combinations produce which effects. To use the table, find the pen color you're using along the left side of the table. Then look on the top edge of the table and find the color of the area on the screen across which you want to draw. Where the row for the pen color and the column for the field color intersect, there's a number. Find the number in the legend to the table and read about the effect you'll get.

About BLACK1, WHITE1, BLACK2, and WHITE2: Due to the vagaries of the Apple's color generation scheme, there are two instances each of the colors black and white. When you look at the Tablet color menu (see the section on PEN COLOR in Chapter 2), you'll see that there are two black squares and two white squares along with the four colored squares. The black and white in the top row are BLACK1 and WHITE1; the ones in the bottom row are BLACK2 and WHITE2. The reason for the duplication is that the 1's cause fewer problems when used with green and violet than do the 2's, and similarly the 2's go better with blue and orange than do the 1's. When this book refers to black or white, it means BLACK1 or WHITE1.

PEN COLOR

	BLACK1	GREEN	VIOLET	WHITE1	BLACK2	ORANGE	BLUE	WHITE2
BLACK1	0	2	2	1	0	2	2	1
GREEN	2	0	3	2	4	5	6	4
VIOLET	2	3	0	2	4	6	5	4
WHITE1	1	2	2	0	1	2	2	0
BLACK2	0	2	2	1	0	2	2	1
ORANGE	4	5	6	4	0	0	3	2
BLUE	4	6	5	4	2	3	0	2
WHITE2	1	2	2	0	1	2	2	0

FIELD COLOR

Color Effects Table

LEGEND:

- 0: No effect.
- 1: Colors appear as expected; no anomalies.
- 2: DASHED LINES on non-horizontal lines; vertical lines may disappear.
- 3: ZEBRA STRIPING on non-horizontal lines; vertical lines appear solid black or white.
- 4: DASHED LINES with a COLOR FLIP.
- 5: Pure COLOR FLIP: non-horizontal lines appear "chunky" and wider than normal.
- 6: ZEBRA STRIPING with a COLOR FLIP.

APPENDIX D PROGRAM LISTINGS

80	Tablet-Code Applesoft
85	Variable Atlas
87	Subroutines
88	Special Locations
89	ROM Code
108	Quick-Draw
118	Utilities

TABLET-CODE APPLESOFT

```

10 REM * TABLET SOFTWARE. COPYRIGHT APPLE 1979. B. EHLERS *
20 LOMEM = 25092
30 B$ = CHR$(34): PRINT D$:"CLOSE GRAPHICS TABLET SOFTWARE"
40 ONERR GOTO 2650
50 B$ = CHR$(14): PRINT D$:"OPEN TAB INFORMATION.D1": PRINT D$:"READ TAB
  INFORMATION": INPUT SL: INPUT XL: INPUT YL: INPUT XH: INPUT YH
60 PRINT: PRINT D$:"CLOSE TAB INFORMATION"
70 ONERR GOTO 2650
80 EP% = PEEK(753) * 256 + PEEK(752): M% = 800
90 DIM Y$(M%): X$(M%)
100 PRINT D$:"LOAD UTILITIES.A56000.D1"
110 XA = YH - XL: YA = YH - YL: LT = INT ((XA + YA) / 2): PI = INT (LT / 11)

120 SO = INT (XA / 11 + 5)
130 MO = INT (PI / 2): XM = XL: YH = 2 + MO + YL
140 HGR2: PC = 0: DC = 0: HCOLOR = PC: W = 1: DF = 1
150 X1 = XM + 2: Y1 = YH + 2: X2 = (INT ((XH + 2 - X1) / 280 + 5) + 280) * Y
  Z = INT (X2 * 192 / 280): X3 = X1 - X2: Y3 = Y1 - Y2
160 D% = -2: SM = 90: GOSUB 1070: RD = 0
170 CM = 0: NV = 1: CALL EP%: CD = PEEK(700): ON CD + 1 GOTO 190,200,170,1
  70
180 GOTO 170
190 PRINT: PRINT D$:"PR#0": PRINT D$:"I#0": GET A$: IF ASC (A$) < 12
  7 THEN 200
192 TEXT: HOME: VTAB 12: HTAB 13: INPUT "QUIT? (Y OR N) ": A$: IF A$ = "
  Y" THEN HOME: VTAB 12: HTAB 10: PRINT "LOADING HELLO PROGRAM": POKE
  104,8: POKE 103,1: PRINT D$:"RUN HELLO.D1": STOP
  GOSUB 1130: GOTO 170
200 IF PEEK(640) < 0 > 2 THEN 170
210 POKE 640,0
220 XF = XL + 2: YF = YL + 2: SF = 50: GOSUB 2390: REM SENSE
  MENU
230 PRINT: PRINT D$:"IN#": SL
240 INPUT X,Y,Z: IF Y > 1 THEN GOSUB 1130: GOTO 170
250 IF Y < 1 AND Y > 0 THEN PRINT D$:"PR#0": PRINT CHR$(7): PRINT
  : PRINT D$:"PR#": SL: PRINT "N": ON Y + 1 GOTO 280,290
260 GOTO 200
270 TEXT: PR# 0: PRINT "ERROR": STOP
280 ON X + 1 GOTO 140,300,690,620,1460,550,1150,2160,1360,650,290,1980,15
  80,1740,1840,570,330,420,2490,2330,1970,2070
290 GOSUB 1130: GOTO 170
300 IF X1 = X3 AND Y1 = Y3 AND X4 = X3 AND Y4 = Y3 THEN HCOLOR = 30: HPL0T
  0,0: CALL 62454: GOTO 520
310 HCOLOR = 30: HPL0T X3,Y3: FOR T1 = Y3 TO Y4: HPL0T X3,T1 TO X4,T1: NEXT
  : HCOLOR = PC: GOSUB 1130: GOTO 170
320 TEXT: PRINT D$:"PR#0": HOME: VTAB 7: HTAB 6: PRINT "PLEASE TYPE THE
  PICTURE NAME ": PRINT: HTAB 7: PRINT D$:"IN#0": INPUT "=> ": A$: IF
  B$ = "" THEN GOSUB 1130: GOTO 170
330 VTAB 9: HTAB 37: CALL - 668: HTAB 1: GOSUB 530
345 B$ = "PIC": B$ = B$ ONERR GOTO 400
350 PRINT D$:"LOAD ".D$: A$4000: V0,D": C$
360 S2 = PEEK(16632) * 256 + PEEK(16633): IF S2 < 0 THEN 130
370 X1 = PEEK(16504) * 256 + PEEK(16505): X2 = PEEK(16506) * 256 + PEEK
  (16507): Y1 = PEEK(16508) * 256 + PEEK(16509): Y2 = PEEK(16510) *
  256 + PEEK(16511)
380 ONERR GOTO 2650
390 HOME: GOTO 520
400 B$ = RIGHT$(B$, (LEN(B$) - 4)): ONERR GOTO 2650
410 PRINT D$:"LOAD ".B$: A$4000: V0,D": C$: GOTO 150
420 TEXT: PRINT D$:"PR#0": HOME: VTAB 7: HTAB 3: PRINT "PLEASE TYPE A N
  AME FOR THIS PICTURE ": PRINT: HTAB 7: PRINT D$:"IN#0": INPUT "=> ":
  B$: IF B$ = "" THEN GOSUB 1130: GOTO 170
430 VTAB 9: HTAB 37: CALL - 668: HTAB 1: GOSUB 530
440 GOSUB 1330: HCOLOR = 30: GOSUB 1040: H = INT (X1 / 256): POKE 16504,H:
  POKE 16505,X1 - H * 256: H = INT (X2 / 256): POKE 16506,H: POKE 1650
  7,X2 - H * 256: H = INT (Y1 / 256): POKE 16508,H: POKE 16509,Y1 - H *
  256
450 H = INT (Y2 / 256): POKE 16510,H: POKE 16511,Y2 - H * 256: H = INT (B
  2 / 256): POKE 16632,H: POKE 16633,S2 - H * 256

```

```

460 B$ = "PIC": B$ = B$ ONERR GOTO 490
470 HCOLOR = PC: PRINT D$:"VERIFY ".B$: A$4000: V0,D": C$: ONERR GOTO 2650
480 VTAB 21: HTAB 1: PRINT "A PICTURE ALREADY EXISTS WITH THAT NAME ": PRINT
  : HTAB 12: INPUT "CONTINUE (Y OR N) ": A$: IF A$ < 0 > "Y" THEN 510
490 ONERR GOTO 2650
500 PRINT D$:"SAVE ".B$: A$4000: L$1FFB: V0,D": C$
510 HOME: PRINT D$:"PR#": SL: PRINT "H2,N"
520 GOSUB 1090: PRINT D$:"IN#": SL: GOTO 170
530 VTAB 10: CALL - 958: PRINT: HTAB 16: PRINT "DRIVE # ? (DEFAULT="
  : DF: "): HTAB 25: INPUT " ": C$: IF C$ < 0 > "1" AND C$ < 0 > "2" THEN
  C$ = STR$(DF)
540 DF = VAL(C$): VTAB 11: HTAB 24: CALL - 958: PRINT C$: RETURN
550 REM *** SOFT RESET COMMAND ***
560 GOSUB 1330: GOSUB 1090: D% = -2: GOTO 170
570 TEXT: PRINT D$:"PR#0": HOME: PRINT D$:"IN#0": GOSUB 530
580 HOME: HTAB 7: PRINT "PRESS SPACE BAR TO CONTINUE ": POKE 34,2
590 PRINT D$:"CATALOG ".D": C$
600 POKE - 16368,0: GET A$: IF A$ < 0 > " " THEN 600
610 GOSUB 1130: GOTO 170
620 REM *** BACKGROUND AND PEN COLOR ***
630 T1 = PC: GOSUB 670: IF PC = 8 THEN PC = T1: GOSUB 1130: GOTO 660
640 BC = PC: PC = T1: HCOLOR = BC: HPL0T 0,0: CALL 62454: GOTO 520
650 T3 = PC: GOSUB 670: GOSUB 1130: IF PC = 8 THEN PC = T3
660 ON CM + 1 GOTO 170,1580,1680,1740,1840
670 XF = XM + 2: YF = YH + 2: SF = INT ((XH - XM) / 140)
680 PRINT D$:"PR#0": TEXT: HOME: PRINT D$:"PR#": SL: PRINT "S1,R,X",XF,0
  : Y,YF,0,SF: GR = HOME: VTAB 22: HTAB 9: PRINT "CONSTRUCTING COLD
  R MENU "
690 COLOR = 5: FOR Z1 = 0 TO 39: HLIN 0,39 AT Z1: NEXT
700 X8 = R: Y8 = 17: X9 = 2: Y9 = 2: C9 = 0: GOSUB 880: X9 = 11: Y9 = 2: C9 = 12
  : GOSUB 880: X9 = 20: Y9 = 2: C9 = 3: GOSUB 880: X9 = 29: Y9 = 2: C9 = 15: GOSUB
  880
710 X9 = 2: Y9 = 21: C9 = 0: GOSUB 880: X9 = 11: Y9 = 21: C9 = 9: GOSUB 880: X9 =
  20: Y9 = 21: C9 = 6: GOSUB 880: X9 = 29: Y9 = 21: C9 = 15: GOSUB 880
720 HOME: VTAB 22: HTAB 7: PRINT "USE THE PEN TO PICK A COLOR ": PRINT:
  : PRINT D$:"PR#": SL: PRINT "N"
730 PRINT D$:"IN#": SL: INPUT X,Y,Z: IF Z < 0 THEN PRINT D$:"IN#0": GET A
  : X = 0: Y = 0: PRINT: IF ASC (A$) = 13 THEN PC = 8: RETURN
  IF Z < 0 > 2 THEN 730
740 X = INT (X / 7): Y = INT (Y / 4)
750 IF Y < 2 OR Y > 37 OR Y = 19 OR Y = 20 OR X < 2 OR X > 37 THEN 730
760 PRINT D$:"PR#0": PRINT CHR$(7): IF Y = 1 AND Y < 19 THEN ON INT (
  (X - 2) / 9) + 1 GOTO 790,800,810,820
770 PRINT D$:"PR#0": PRINT CHR$(7): IF Y = 1 AND Y < 19 THEN ON INT (
  (X - 2) / 9) + 1 GOTO 930,940,950,960
780 ON INT ((X - 2) / 9) + 1 GOTO 930,940,950,960
790 PC = 0: B$ = "BLACK1": GOTO 870
800 PC = 1: B$ = "GREEN": GOTO 870
810 PC = 2: B$ = "VIOLET": GOTO 870
820 PC = 3: B$ = "WHITE1": GOTO 870
830 PC = 4: B$ = "BLACK2": GOTO 870
840 PC = 5: B$ = "ORANGE": GOTO 870
850 PC = 6: B$ = "BLUE": GOTO 870
860 PC = 7: B$ = "WHITE2"
870 HOME: VTAB 22: HTAB (40 - LEN(B$)) / 2: PRINT B$: FOR Z1 = 1 TO 50
  0: NEXT: HCOLOR = PC: HOME: RETURN
880 COLOR = C9: FOR Z1 = 1 TO X8: VLIN Y9,Y9 + Y8 - 1 AT X9: X9 = X9 + 1: NEXT
  : RETURN: REM COLOR BOX LO-RES DRAW
890 REM *** WINDOW COMMAND ***
900 PRINT: PRINT D$:"PR#": SL: PRINT "T1,F,C": PRINT D$:"PR#0"
910 TEXT: HOME: VTAB 9: HTAB 15: PRINT "PRESS PEN AT": PRINT: HTAB 8: INVERSE
  PRINT D$:"PR#": SL: PRINT "N,C": PRINT D$:"IN#": SL: INPUT X,Y,Z: IF Z <
  0 THEN PRINT D$:"IN#0": GET A$: IF ASC (A$) = 68 THEN GOSUB 1330: HCOLOR =
  80: GOSUB
  1040: X1 = X3: X2 = X6: Y1 = Y5: Y2 = Y6: GOSUB 1070: GOTO 170
  IF Z < 0 THEN IF ASC (A$) = 13 THEN GOSUB 1130: GOTO 170
  IF Z < 0 > 2 THEN 920
920 PRINT D$:"PR#0": IF X < XM + 2 OR Y < YH + 2 THEN VTAB 22: HTAB 4: PRINT
  CHR$(7): "PLEASE STAY WITHIN THE WORK-AREA ": FOR Z1 = 1 TO 500: NEXT
  Z1: VTAB 22: CALL - 668: GOTO 920
930 VTAB 11: CALL - 668: HTAB 3: PRINT "UPPER-LEFT AND ": INVERSE: PRINT
  "LOWER-RIGHT": NORMAL
940 PRINT D$:"PR#": SL: PRINT "N,C": INPUT TX,TY,Z: IF Z < 0 THEN PRINT D
  $:"IN#0": GET A$: IF ASC (A$) = 68 THEN GOSUB 1330: HCOLOR = 80: GOSUB
  1040: X1 = X3: X2 = X6: Y1 = Y5: Y2 = Y6: GOSUB 1070: GOTO 170
  IF Z < 0 THEN IF ASC (A$) = 13 THEN GOSUB 1130: GOTO 170
  IF Z < 0 > 2 THEN 970
970

```



```

1000 PRINT PRINT D$;"PR#0": IF TX > XH * 2 OR TY > YH * 2 THEN VTAB 22
    HTAB 4: PRINT CHR$(7): "PLEASE STAY WITHIN THE WORK AREA." FOR ZZ
    = 1 TO 500: NEXT ZZ: VTAB 22: CALL - 868: GOTO 970
1010 IF TX < Y OR TY < Y THEN VTAB 22: HTAB 5: PRINT CHR$(7): "PLEASE S
    PECIFY POINTS CORRECTLY!": FOR ZZ = 1 TO 500: NEXT ZZ: GOTO 910
1020 X1 = X: X2 = TX - X1 + 1: Y1 = Y: Y2 = TY - Y1 + 1
1030 GOSUB 1330: HCOLOR = BC: GOSUB 1040: GOSUB 1070: GOTO 170
1040 XT = INT (GF): XB = 279 - XT: YT = INT (HF): YB = 191 - YT: IF XT >
    = 2 THEN HPLLOT XT - 1, YT TO XT - 1, YB: HPLLOT XT - 2, YT TO XT - 2, YB: HPLLOT
    XB + 1, YT TO XB + 1, YB: HPLLOT XB + 2, YT TO XB + 2, YB
1050 IF YT > = 1 THEN HPLLOT XT, YT - 1 TO XB, YT - 1: HPLLOT XT, YB + 1 TO
    XB, YB + 1
1060 RETURN
1070 T1 = X2 / 280: T2 = Y2 / 192: IF T1 < T2 THEN S2 = T2: IF INT (T2) <
    T2 THEN S2 = INT (T2) + 1
1080 IF T1 > = T2 THEN S2 = T1: IF INT (T1) < T1 THEN S2 = INT (T1) +
    1
1090 RD = 0: GOSUB 1130: HCOLOR = 0: IF BC = 0 OR BC = 4 THEN HCOLOR = 3
1100 GOSUB 1040: HCOLOR = PC: B1 = INT (XT / 256): B2 = XT - B1 * 256: B3 =
    INT ((XB + 1) / 256): B4 = (XB + 1) - B3 * 256: B5 = YT - B6 * 256 + 1: X3
    = INT (XT) * X4 = INT (XB) * Y3 = INT (YT) * Y4 = INT (YB)
1110 GOSUB 1330: HCOLOR = PC: WH = 1: WS = 1: ONERR GOTO 2650
1120 BX = 3089: POKE BX, B2: POKE BX + 1, B1: POKE BX + 2, B4: POKE BX + 3, B3
    POKE BX + 4, B5: POKE BX + 5, B6: RETURN
1130 IF RD > 0 THEN GOSUB 1400: RETURN
1140 GF = (280 - X2 / S2) / 2: HF = (192 - Y2 / S2) / 2: XF = INT (X1 + GF *
    S2): YF = INT (Y1 - HF * S2): SF = S2: GOSUB 2590: RETURN
1150 REM *** VIEWPORT COMMAND ***
1160 HCOLOR = 3: IF BC = 3 OR BC = 7 THEN HCOLOR = 0
1170 GOSUB 1330: XF = INT (X1 - GF * S2): YF = INT (Y1 - HF * S2): SF = S2
    GOSUB 2590
1180 GOSUB 1290: GOSUB 1310
1190 IF Z < 0 THEN PRINT D$;"IN#0": PRINT : GET A$: IF ASC (A$) = 13 THEN
    GOSUB 1130: GOSUB 1330: HCOLOR = PC: GOTO 170
1200 IF Z < 0 THEN IF ASC (A$) = 68 THEN GOSUB 1090: GOTO 170
1205 IF Z < 0 THEN GOTO 1180
1210 IF X > X OR XB < X OR YT > Y OR YB < Y THEN 1180
1220 T1 = X - 1: T2 = Y - 1: H = 0: XB = T1: YB = T2: GOSUB 1350
1230 GOSUB 1280: GOSUB 1310: IF Z < 0 THEN GOSUB 1350: GOTO 1190
1240 IF XT > Y OR XB < X OR YT > Y OR YB < Y THEN 1230
1250 IF X < T1 OR Y < T2 THEN GOSUB 1350: PRINT D$;"PR#0": TEXT - HOME
    VTAB 12: HTAB 5: PRINT "PLEASE SPECIFY POINTS CORRECTLY": GOSUB 1300
    GOSUB 1140: GOTO 1180
1260 WH = 1: WS = 1: GOSUB 2590: RD = 0
1270 X2 = T1 + 1: Y2 = T2 + 1: X4 = X: Y4 = Y: H = 0: GOSUB 1350: GOSUB 1330: B
    1 = INT (X3 / 256): B2 = X3 - B1 * 256: B3 = INT ((X4 + 1) / 256): B4 =
    (X4 + 1) - B3 * 256: B5 = Y3 - B6 * 256 + 1: GOSUB 1120: HCOLOR = PC: GOTO
    170
1280 PRINT PRINT D$;"PR#0": GOSUB 2300: PRINT D$;"PR#": SL: PRINT "M2": VTAB
    23: HTAB 15: POKE 41: PEEK (41) + 4: PRINT CHR$(7): "LOWER-RIGHT?": FOR
    T3 = 1 TO 500: NEXT T3: PRINT D$;"PR#": SL: PRINT "N.H2": RETURN
1290 PRINT PRINT D$;"PR#0": GOSUB 2300: PRINT D$;"PR#": SL: PRINT "M2": VTAB
    23: HTAB 15: POKE 41: PEEK (41) + 4: PRINT CHR$(7): "UPPER-LEFT?": FOR
    T3 = 1 TO 500: NEXT T3: PRINT D$;"PR#": SL: PRINT "N.H2": RETURN
1300 FOR H = 1 TO 1000: NEXT H: RETURN
1310 PRINT D$;"IN#": SL: INPUT "X, Y, Z: IF Z = 2 OR Z < 0 THEN RETURN
1320 GOTO 1310
1330 POKE 233, 99: POKE 232, 32: HCOLOR = 0: IF BC = 0 OR BC = 4 THEN HCOLOR =
    3
1340 H = 0: YB = X3 - 1: YB = Y3 - 1: GOSUB 1350: H = 16: XB = X4 + 1: YB = Y3 -
    1: GOSUB 1350: H = 32: XB = Y4 + 1: YB = Y4 + 1: GOSUB 1350: H = 48: XB =
    X3 - 1: YB = Y4 + 1: GOSUB 1350: RETURN
1350 IF XB > = 0 AND XB < 280 AND YB > = 0 AND YB < 192 THEN ROT = H: SCALE =
    1: XDRAW 1 AT XB, YB
1360 RETURN
1380 RD = RD + 1: IF RD > 1 THEN RD = 0: GOSUB 1130: ON CM + 1 GOTO 170, 15
    80, 1680, 1740, 1840
1390 GOSUB 1400: ON CM + 1 GOTO 170, 1580, 1680, 1740, 1840
1400 IF X4 = X3 OR Y4 = Y3 THEN 1440
1410 T1 = ((XH * 2) - (XM * 2)) / (X4 - X3): T2 = ((YH * 2) - (YM * 2)) / (Y
    4 - Y3): SF = INT (T1): IF T2 < T1 THEN SF = INT (T2)
1420 XF = INT ((XM * 2) - (SF * X3)): YF = INT ((YM * 2) - (SF * Y3)): IF
    ABS (XF) > 27000 OR ABS (YF) > 27000 THEN GOTO 1440

```

```

1430 GOSUB 2590: RETURN
1440 PRINT D$;"PR#0": GOSUB 2300: PRINT : PRINT D$;"PR#": SL: PRINT "M2.C"
    VTAB 23: HTAB 14: POKE 41: PEEK (41) + 4: PRINT "NOT POSSIBLE": RD =
    0: GOSUB 1300: GOSUB 1130: RETURN
1450 PRINT : PRINT D$;"PR#": SL: PRINT "T1.C": PRINT D$;"PR#0": PRINT D$;"
    IN#0": TEXT
1470 HOME: PRINT : HTAB 9: PRINT "FAST-DRAW DELTA SETTING": VTAB 5: HTAB
    7: PRINT "CURRENT DELTA SETTING IS ": ABS (D$): " ": PRINT A$ = "ON."
    IF D$ < 0 THEN A$ = "OFF."
1480 HTAB 10: PRINT "AUDIO FEEDBACK IS ": A$
1490 VTAB 18: CALL - 958: HTAB 11: INPUT "NEW DELTA EQUALS ": A$: IF A$ =
    "" THEN 1530
1500 IF VAL (A$) < 1 OR VAL (A$) > 127 THEN 1490
1510 IF D$ < 1 THEN D$ = - VAL (A$): GOTO 1520
1520 D$ = VAL (A$)
1530 VTAB 20: CALL - 958: HTAB 9: INPUT "TURN AUDIO FEEDBACK ": A$: IF A$
    = "" THEN 1560
1540 IF LEFTS (A$, 2) < 0 "ON" AND LEFTS (A$, 3) < 0 "OFF" THEN 1530
1550 D$ = ABS (D$): IF LEFTS (A$, 3) = "OFF" THEN D$ = - D$
1560 GOSUB 1130: GOTO 170
1580 GOSUB 1130: PRINT D$;"IN#": SL: CM = 1
1590 RT = 2: INPUT X, Y, Z: IF Z < 0 > 2 THEN POKE - 16368, 0: GOTO 1590
1600 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1610 IF RT = 0 THEN 1590
1620 HPLLOT X, Y
1630 RT = 2: INPUT X, Y, Z: IF Z < 0 > 2 THEN POKE - 16368, 0: GOTO 1630
1640 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1650 IF RT = 0 THEN 1630
1660 HPLLOT TO X, Y: GOTO 1630
1680 GOSUB 1130: PRINT D$;"IN#": SL: CM = 2
1690 RT = 2: INPUT X, Y, Z: IF Z < 0 > 2 THEN POKE - 16368, 0: GOTO 1690
1700 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1710 IF RT = 0 THEN 1690
1720 HPLLOT X, Y: GOTO 1690
1740 GOSUB 1130: PRINT D$;"IN#": SL: CM = 3
1750 RT = 2: INPUT X, Y, Z: IF Z < 0 > 2 THEN POKE - 16368, 0: GOTO 1750
1760 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1770 IF RT = 0 THEN 1750
1780 HPLLOT X, Y: TX = X: TY = Y
1790 RT = 2: INPUT X, Y, Z: IF Z < 0 > 2 THEN POKE - 16368, 0: GOTO 1790
1800 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1810 IF RT = 0 THEN 1790
1820 HPLLOT X, Y TO TX, Y TO TX, TY TO X, TY TO X, Y: GOTO 1750
1840 GOSUB 1130: PRINT D$;"IN#": SL: CM = 4
1850 RT = 2: INPUT X, Y, Z: IF Z < 0 > 2 THEN POKE - 16368, 0: GOTO 1850
1860 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1870 IF RT = 0 THEN 1850
1880 HPLLOT X, Y: TX = X: TY = Y
1890 RT = 2: INPUT X, Y, Z: IF Z < 0 > 2 THEN POKE - 16368, 0: GOTO 1890
1900 IF X < X3 OR X > X4 OR Y < Y3 OR Y > Y4 THEN GOSUB 1940: IF RT = 1 THEN
    220
1910 IF RT = 0 THEN 1890
1920 IF Y < TY THEN FOR H = Y TO TY: HPLLOT X, H TO TX, H: NEXT H: GOTO 1850
1930 FOR H = TY TO Y: HPLLOT X, H TO TX, H: NEXT H: GOTO 1850
1940 IF (Y = SF * YF - YL * 2) / 50 < 2 THEN RT = 1: RETURN
1950 PRINT D$;"PR#0": GOSUB 2300: PRINT D$;"PR#": SL: PRINT "M2": VTAB 23:
    HTAB 3: POKE 41: PEEK (41) + 4: PRINT "POINT OUTSIDE VIEWPORT: RESP
    ECIFY.": GOSUB 1300: PRINT D$;"PR#": SL: PRINT "N.H2": RT = 0: RETURN
1970 GOSUB 1130: X$ = 1: CALL EPI: CD = PEEK (700): ON CD + 1 GOTO 190, 197
    5, 1980, 1980
1975 IF N$ = 1 THEN 1970
1980 HPLLOT X$(1), Y$(1) TO X$(N$ - 1), Y$(N$ - 1): GOSUB 1990: GOSUB 1130: GOTO
    170
1990 PRINT D$;"PR#0": GOSUB 2300: PRINT D$;"PR#": SL: PRINT "M2": VTAB 23:
    HTAB 14: POKE 41: PEEK (41) + 4: PRINT CHR$(7): "CALCULATING. ": IF
    N$ = 2 THEN AR = 0: GOTO 2020

```



```

2000 AR = 0: FOR T1 = 2 TO NX - 1: DX = X%(T1) - X%(T1 - 1): DY = Y%(T1) - Y%(T1 - 1): 2: AR = AR + DX * DY: NEXT T1
2010 AR = AR + (X%(1) - X%(NX - 1)) * (Y%(1) + Y%(NX - 1)) / 2: AR = ABS(AR) / 4: 2: IF AR < 99999999 THEN AR = (INT (AR * 100)) / 100
2020 GOSUB 2300: VTAB 23: B$ = "AREA IS ": POKE 41: PEEK (41) + 4: GOSUB 2300: GOSUB 1300: GOSUB 1300: RETURN
2030 B$ = B$ + STR$(AR) + " SQUARE " + W$ + " ": HTAB 21: INT (LEN (B$) / 2): PRINT B$: RETURN
2040 GOSUB 1130: NX = 1: CALL EP%: CD = PEEK (1700): ON CD + 1 GOTO 170, 207, 5, 2080, 2090
2075 IF NX = 1 THEN 2070
2080 GOSUB 2090: GOSUB 1130: GOTO 170
2090 PRINT D$:"PR#0": GOSUB 2300: PRINT D$:"PR#": SL: PRINT "M2": VTAB 23: HTAB 14: POKE 41: PEEK (41) + 4: PRINT CHR$(7): "CALCULATING. " IF NX = 2 THEN DT = 0: GOTO 2110
2100 DT = 0: FOR T1 = 2 TO NX - 1: DX = X%(T1) - X%(T1 - 1): DY = Y%(T1) - Y%(T1 - 1): DT = DT + SQR (DX * DX + DY * DY): NEXT T1: DT = DT / 4: IF DT < 99999999 THEN DT = (INT (DT * 100)) / 100
2110 GOSUB 2300: VTAB 23: B$ = "THE DISTANCE IS ": POKE 41: PEEK (41) + 4: GOSUB 1130: GOSUB 1300: GOSUB 1300: RETURN
2120 B$ = B$ + STR$(DT) + " " + W$ + " ": HTAB 21: INT (LEN (B$) / 2): PRINT B$: RETURN
2130 GOSUB 1130
2170 GOSUB 2310
2180 PRINT D$:"IN#": SL: INPUT X,Y,Z: IF Z < 0 THEN PRINT D$:"IN#0": GET A$: IF ASC (A$) = 13 THEN GOSUB 1130: GOTO 170
2190 IF Z < 0: 2 THEN PRINT: GOTO 2180
2200 IF X < 0: X OR X4 < 0: OR Y3 < 0: OR Y4 < 0: Y THEN 2170
2210 GOSUB 2320
2220 PRINT D$:"IN#": SL: INPUT TX,TY,Z: IF Z < 0 THEN PRINT D$:"IN#0": GET A$: IF ASC (A$) = 13 THEN GOSUB 1130: GOTO 170
2230 IF Z < 0: 2 THEN PRINT: GOTO 2220
2240 IF TX < 0: Y3 OR TX > 0: X4 OR TY < 0: Y3 OR TY > 0: Y4 THEN 2210
2250 PRINT D$:"PR#": SL: PRINT "T1,C": TEXT: HOME: T1 = TX - X + 1: T2 = TY - Y + 1: DX = SQR (T1 * T1 + T2 * T2): VTAB 10: HTAB 6: PRINT "DISTANCE IS ": INT (DX): SCREEN UNITS: PRINT D$:"IN#0"
2260 VTAB 18: CALL - 958: HTAB 8: INPUT "YOUR NUMBER OF UNITS -> ": A$: IF A$ = "" THEN W = DX: GOTO 2260
2265 IF VAL (A$) > 99999999 THEN 2260
2270 W = VAL (A$): IF W = 0 THEN 2260
2280 VTAB 20: CALL - 958: HTAB 8: INPUT "TYPE OF UNITS -> ": W$: IF LEN (W$) > 10 THEN 2280
2290 WM = DX / W: GOSUB 1130: GOTO 170
2300 FOR T4 = 21 TO 24: VTAB T4: HTAB 1: POKE 41: PEEK (41) + 4: PRINT " ": NEXT T4: PRINT: RETURN
2310 PRINT D$:"PR#0": GOSUB 2300: PRINT D$:"PR#": SL: PRINT "M2": VTAB 23: HTAB 13: POKE 41: PEEK (41) + 4: PRINT CHR$(7): "BEGINNING POINT"
2320 FOR T3 = 1 TO 500: NEXT: PRINT D$:"PR#": SL: PRINT "M2": RETURN
2320 PRINT D$:"PR#0": GOSUB 2300: PRINT D$:"PR#": SL: PRINT "M2": VTAB 23: HTAB 14: POKE 41: PEEK (41) + 4: PRINT CHR$(7): "ENDING POINT"
2330 FOR T3 = 1 TO 500: NEXT: PRINT D$:"PR#": SL: PRINT "M2": RETURN
2330 REM *** CHRIS' SLIDE ***
2340 GOSUB 1300: HCOLOR= 80: GOSUB 1040
2350 GOSUB 1130: GOSUB 2310
2360 PRINT D$:"IN#": SL: INPUT X,Y,Z: IF Z < 0 THEN PRINT D$:"IN#0": GET A$: IF ASC (A$) = 13 THEN 2470
2370 IF Z < 0: 2 THEN PRINT: GOTO 2360
2380 IF X < 0: OR X > 279: OR Y < 0: OR Y > 191: THEN 2330
2390 GOSUB 2320
2400 PRINT D$:"IN#": SL: INPUT TX,TY,Z: IF Z < 0 THEN PRINT D$:"IN#0": GET A$: IF ASC (A$) = 13 THEN 2470
2410 IF Z < 0: 2 THEN PRINT: GOTO 2400
2420 IF TX < 0: OR TX > 279: OR TY < 0: OR TY > 191: OR (TX = X AND TY = Y) THEN 2390
2430 IF TY > Y THEN FOR Z2 = 1 TO TY - Y: CALL 25218: NEXT: GOTO 2430
2440 IF Y > TY THEN FOR Z2 = 1 TO Y - TY: CALL 25175: NEXT
2450 IF TX > X THEN FOR Z2 = 1 TO INT ((TX - X) / 14): CALL 25308: NEXT: GOTO 2470
2460 IF X > TX THEN FOR Z2 = 1 TO INT ((X - TX) / 14): CALL 25261: NEXT
2470 HCOLOR= 0: IF BC = 0 OR BC = 4 THEN HCOLOR= 3
2480 GOSUB 1040: GOSUB 1330: HCOLOR= PC: GOSUB 1130: GOTO 170
2490 REM *** DAVE'S SEPERATE ***

```

```

2500 T3 = PC: GOSUB 670: IF PC = 4 OR PC = 0 THEN PC = T3: HCOLOR= PC: HOME: PRINT: HTAB 9: PRINT "NO SEPARATION ON SLABS.": GOSUB 1300: GOSUB 1410: GOSUB 1130: GOTO 170
2510 IF PC = 8 THEN PC = T3: GOTO 2580
2520 PRINT D$:"PR#": SL: PRINT "M2,N": GOSUB 1330: HCOLOR= 80: GOSUB 1040: IF PC = 3 OR PC = 7 THEN POKE 767,176: CALL 24576: GOTO 2570
2530 POKE 767,144: CALL 24576: T1 = 128: T2 = 213: IF PC = 2 THEN T2 = 170
2540 IF PC = 5 THEN T1 = 0: T2 = 213
2550 IF PC = 6 THEN T1 = 0: T2 = 170
2560 POKE 766,T1: POKE 767,T2: CALL 24911
2570 BC = 0: HCOLOR= 3: GOSUB 1040: GOSUB 1330
2580 HCOLOR= PC: GOSUB 1130: GOTO 170
2590 PRINT: PRINT D$:"PR#": SL
2600 PRINT "D,S, SF, M2, X, XF, Y, YF, R,N": RETURN
2610 TEXT: HOME: PRINT: HTAB 7: PRINT "TABLET INFORMATION FILE DOES" PRINT: HTAB 15: PRINT "NOT EXIST"
2620 VTAB 7: HTAB 8: PRINT "MAKE SURE THE MASTER DISK" PRINT: HTAB 11: PRINT "IS NOT PROTECTED AND" PRINT: HTAB 12: PRINT "THEN PRESS RETURN"
2630 VTAB 14: HTAB 5: PRINT "THE MENU ALIGNMENT ROUTINE WILL" PRINT: HTAB 17: PRINT "BE RUN. ": GET A$: IF ASC (A$) < 13 THEN 2630
2635 POKE 104,8: POKE 103,1
2640 PRINT: PRINT 38: "RUN MENU ALIGNMENT.D1" STOP
2650 REM * ERROR HANDLER *
2660 TEXT: HOME: T7 = PEEK (222): PRINT D$:"PR#0": IF T7 = 3 THEN VTAB 12: HTAB 16: PRINT "I/O ERROR.": GOTO 2700
2670 IF T7 = 4 THEN VTAB 12: HTAB 11: PRINT "PICTURE NOT ON DISK.": GOTO 2700
2680 IF T7 = 4 OR T7 = 9 OR T7 = 10 THEN VTAB 12: HTAB 8: PRINT "THE PICTURE IS LOCKED. OR": HTAB 5: PRINT "THE DISK IS FULL, OR PROTECTED.": GOTO 2700
2685 IF T7 = 13 THEN VTAB 12: PRINT "FILE REQUESTED IS NOT A PICTURE FILE.": GOTO 2700
2690 VTAB 12: HTAB 9: PRINT "PROBLEM ->": PEEK (222) = T7
2700 VTAB 20: HTAB 8: PRINT "PRESS SPACE BAR TO RETRY.": PRINT: HTAB 11: PRINT "PRESS CCRD TO ADDIT"
2710 VTAB 24: HTAB 20: GET A$: IF A$ = " " THEN VTAB 20: HTAB 1: CALL - 958: HTAB 15: PRINT "RETRYING.": IF T7 = 6 THEN GOTO 345
2715 IF A$ = " " THEN RESUME
2720 IF ASC (A$) = 13 THEN PRINT: PRINT D$:"CLOSE ": B$: GOTO 290
2730 GOTO 2710

```

VARIABLE ATLAS

Name	Description
A\$	General-purpose input string
AR	Calculated area for AREA command
B\$	Input string for picture name in LOAD and SAVE
B1-B6	Temporary variables for WINDOW
BC	Background Color (defaults to 0)
BX	Pointer into the QUICK-DRAW subroutines
C\$	String for slot number
CD	Termination code of QUICK-DRAW subroutines
CM	Current Command mode: 0 = DRAW 1 = LINES 2 = DOTS 3 = FRAME 4 = BOX
D\$	CTRL-D (CHR\$(4)) for DOS commands
DZ	DELTA setting (0-127; negative if Audio Feedback is off)
DP	Default drive number for LOAD, SAVE, CATALOG

DT Calculated distance for DISTANCE command

DX,DY Temporary variables used in AREA and DISTANCE: the vertical and horizontal distance between a point and the next one.

E\$ Temporary input string for SAVE

EP% The beginning address of the QUICK-DRAW subroutines

GF X screen offset values for WINDOW

H Widely used as a temporary variable.

HF Y screen offset values for WINDOW

LT Length of menu, in Tablet units

M% Maximum number of points for DISTANCE or AREA calculation

MD Height and width of each menu command square, in Tablet units

N% Index into arrays X% and Y%, used by QUICK-DRAW

PC Pen color (0-7), defaults to 3 (white)

PI Number of points per inch on the Tablet

RD Flag for REDUCER mode: 1=on, 0=off.

RT A return flag for LINES, DOTS, FRAME, and BOX modes whose value indicates the phase of the operation:

0 = Operation was just initialized.

1 = Menu selected; operation cancelled.

2 = Operation in progress.

S0 Scale setting for menu

S2 Scale setting for WINDOW after LOAD

SF Scale Factor -- see XF,YF

SL Slot number of Tablet Interface card (read from info file)

T1-T9 Temporary variables

TX,TY Temporarily holds an X,Y position (for BOX, LINES, FRAME, SLIDE)

W User CALIBRATE units

W\$ Name of user CALIBRATE units

WM CALIBRATE multiplier (WM= Tablet units / W)

X,Y General-purpose coordinate pair for high-resolution screen

X%,Y% Arrays (of length M%) which hold coordinates of points plotted in DRAW, AREA, and DISTANCE. They are filled by the QUICK-DRAW subroutines.

X1,Y1 Coordinates for upper-left corner of WINDOW on Tablet

X2,Y2 Coordinates for lower-right corner of WINDOW on Tablet

X3,Y3 Coordinates for upper-left corner of VIEWPORT on screen

X4,Y4 Coordinates for lower-right corner of VIEWPORT on screen

X5,Y5 Default values for X1,Y1

X6,Y6 Default values for X2,Y2

X8,Y8 Temporary X,Y coordinates (for VIEWPORT and color menu)

X9,Y9 " " " "

XA,YA Width and height of menu overlay

XB,YB Coordinates for lower-right corner of WINDOW on screen

XF,YF Current Tablet offset factors

XH,YH Coordinates for upper-left corner of overlay on Tablet

XL,YL Coordinates for lower-right corner of overlay on Tablet

XM,YM Coordinates for upper-left corner of working area on Tablet

XT,YT Coordinates for upper-left corner of WINDOW on screen

Z Pen up/pen down value:

0 = pen is down, and has been down.

1 = pen is up

2 = pen newly down

10 = pen is off-scale

Negative numbers indicate that a key has been pressed.

ZZ Temporary variable used in delay loops.

SUBROUTINES

Entry	Description
530	Inputs drive number from keyboard
670	Displays color menu; returns chosen color in PC
880	Draws a single box of the color C9 on the low-resolution graphics screen. The box will be X8 blocks tall, and its upper-left corner will be at (X9, Y9).
1040	Draws the WINDOW on the high-resolution screen in the current HCOLOR.
1070	Sets scaling information for Tablet; falls into subroutine at 1090
1090	Turns off REDUCER, removes WINDOW frame and sets WINDOW to its default values, resets CALIBRATE setting, and falls into subroutine at 1120
1120	Stores VIEWPORT setting in memory for QUICK-DRAW
1130	Resets Tablet scaling information (with REDUCER, if active)
1280	Prints prompt "LOWER-RIGHT?"
1290	Prints prompt "UPPER-LEFT?"
1300	Delay 1.1 seconds
1310	Wait for the pen to be down or a keypress. If pen is down, return with coordinates in X,Y; if keypress, return with Z<0.
1330	Draws or undraws the four VIEWPORT corner marks.
1350	Draws or undraws a single VIEWPORT corner mark. The corner's coordinates are in X8,Y8 and the rotation factor is in H.
1400	Turns on the REDUCER.
1940	Returns with RT=1 if the last pen press was in the menu area; otherwise displays "POINT OUTSIDE VIEWPORT. RESPECIFY"
1990	Performs an AREA calculation on the polygon whose vertices are in the arrays X%,Y%. Returns with the area in AR.
2090	Performs a DISTANCE calculation on the closed curve whose points are in the arrays X%,Y%. Returns with the distance in DT.
2120	Adds the value of DT to the end of string B\$, and prints it centered on the screen.
2300	Clears out the bottom four lines of the Page 2 Text screen.
2310	Displays the prompt "BEGINNING POINT?"
2320	Displays the prompt "ENDING POINT?"
2590	Reinitializes the Tablet with the scaling factor in SF, the X-offset in XF, and the Y-offset in YF.

SPECIAL LOCATIONS

These special memory locations are used by the TABLET-CODE APPLESOFT program. The decimal addresses are given on the left; hexadecimal equivalents are in parentheses and preceded by a dollar sign (\$):

Location	Use
41 (\$29)	This location contains the high part of the memory address of the beginning of the current line on the Text screen. A POKE 41, PEEK(41)+4 operation will cause the next printed line to appear on Page 2, rather than Page 1, of Text mode.
103,104 (\$67,\$68)	This pair of locations holds the address of the beginning of the current Applesoft program in memory.
222 (\$DE)	This location holds the ON ERR GOTO code of the last error generated.
232,233 (\$E8,\$E9)	This pair of locations holds the address of the beginning of the current shape table for the Applesoft DRAW and XDRAW commands.
700 (\$2BC)	Holds the termination code from the QUICK-DRAW subroutines.
752,753 (\$2F0,\$2F1)	After the QUICK-DRAW program is RUN, this pair of locations will hold the memory address of the beginning of the QUICK-DRAW subroutine.
766,767 (\$2FE,\$2FF)	These locations are used to pass the selected color to the SEPARATE subroutine.
3089-3094 (\$C11-\$C16)	These locations are used to pass VIEWPORT information to the QUICK-DRAW subroutine.
16632,16633 (\$40F8,\$40F9)	These locations are in the memory range used by the high-resolution graphics Page 2, but their contents are neither displayed on the screen or affected by normal screen operations. These two locations are used to store the value of S2 during a SAVE.
16504-16511 (\$4078-\$407F)	These are also locations in the high-resolution Page 2 which are not displayed. These eight locations are used to store the values of X1, X2, Y1, and Y2 during a SAVE.
24576 (\$6000)	This is the entry point for the machine language subroutine which performs a SEPARATE.

24911 (\$614F)	This is another entry point for SEPARATE.
25175 (\$6257)	Entry point for a one-dot SLIDE down.
25218 (\$6282)	Entry point for a one-dot SLIDE up.
25261 (\$62AD)	Entry point for a 14-dot SLIDE right.
25308 (\$62DC)	Entry point for a 14-dot SLIDE left.
-16368 (\$C010)	A PEEK or POKE to this location will clear the Apple's keyboard strobe, causing any recent keypress to be ignored.
62454 (\$F3F6)	This subroutine in the Applesoft ROM fills the entire high-resolution screen with the most recent HCOLOR plotted.
-958 (\$FC42)	This subroutine in the Apple's Monitor ROM clears the text screen from the current cursor position to the end of the screen.
-868 (\$FC9C)	This subroutine in the Apple's Monitor ROM clears the text screen from the current cursor position to the end of the line.

ROM CODE

```

SOURCE FILE: BITPAD35 1F
SOURCE FILE: BITPAD35 2F
0000      1 *****
0000      2 *
0000      3 *
0000      4 *      BIT PAD FIRMWARE
0000      5 *
0000      6 *      COPYRIGHT APPLE COMPUTER
0000      7 *      7/30/79
0000      8 *      W SANDER
0000      9 *
0000     10 *
0000     11 *****
0024     12 CH      EQU    $24      ; SCREEN HORIZONTAL POSITION
002A     13 KBASL   EQU    $2A      ; BASE ADDRESS FOR BITPAD CURSOR
002B     14 HDASH   EQU    $2B
002B     15 BASL    EQU    $2B      ; TEXT BASE ADDRESS
0036     16 COUTL   EQU    $36      ; LOW BYTE OF COUT POINTER
0037     17 COUTH   EQU    $37      ; HIGH BYTE OF COUT POINTER
0200     18 INQ     EQU    $200     ; INPUT BUFFER ADDRESSES
0201     19 IN1     EQU    $201
0202     20 IN2     EQU    $202
0203     21 IN3     EQU    $203
0280     22 TEM     EQU    $280      ; RETURN FLAG LOCATION
0000     23 *      ; HIGH NIBBLE - 1=RETURN SCALED VALUE
0000     24 *      ; LOW NIBBLE - 0=PEN DOWN, 1=PEN LEFT, 2=PEN FALL, 3=PEN UP
0281     25 XFLL    EQU    $281      ; X-COORD LOW BYTE, FULL SCALE
0282     26 XFLH    EQU    $282      ; X-COORD HIGH BYTE, FULL SCALE
0283     27 YFLL    EQU    $283      ; Y-COORD LOW BYTE, FULL SCALE
0284     28 YFLH    EQU    $284      ; Y-COORD HIGH BYTE, FULL SCALE
0285     29 TEMXL   EQU    $285      ; X-COORD LOW BYTE, SCALED
0286     30 TEMX    EQU    $286      ; X-COORD HIGH BYTE, SCALED

```



```

0287: 31 TEMYL EQU $287 ;Y-COORD LOW BYTE, SCALED
0288: 32 TEMY EQU $288 ;Y-COORD HIGH BYTE, SCALED
0289: 33 REGL EQU $289 ;DIVIDE REGISTERS
0290: 34 REGH EQU $289
0291: 35 INA EQU $290 ;BUFFER REGISTER FOR PR# SYNTAX
0292: 36 INX EQU $291 ;PR# BUFFER POINTER
0293: 37 NFLAG EQU $292 ;PR# BUFFER STATUS FLAG
0294: 38 SAVSLOT EQU $293
02A0: 39 MIFLAG EQU $2A0
02A1: 40 OREGL EQU $2A1
02A2: 41 OREGH EQU $2A2
02A3: 42 DIVL EQU $2A3
02A4: 43 DIVH EQU $2A4
02A5: 44 C1HAR EQU $2A5
02B8: 45 PAGE EQU $2B8 ;PAGE CODE
0000: 46 *
0000: 47 * HIGH BIT = 1 MEANS SCALE DATA
0000: 48 * 40 = HIRES PAGE2
0000: 49 * 20 = HIRES PAGE1
0000: 50 * 08 = TEXT PAGE2
0000: 51 * 04 = TEXT PAGE1
0000: 52 * 02 = LORES PAGE2
0000: 53 * 01 = LORES PAGE1
0000: 54 * 42 = HIRES MIXED PAGE2
0000: 55 * 21 = HIRES MIXED PAGE1
0000: 56 * 0A = LORES MIXED PAGE2
0000: 57 * 05 = LORES MIXED PAGE1
0000: 58 *
0438: 59 MPAGE EQU $438
0000: 60 *
0000: 61 * LAST SIX BITS OF MPAGE CORRESPOND TO PAGE
0000: 62 * BIT 7 MEANS STREAM MODE IF 1
0000: 63 * BIT 6 MEANS OFFSET AFTER SCALE IF 1
0000: 64 *
04B8: 65 SCALL EQU $4B8 ;LOW BYTE OF SCALE FACTOR
0538: 66 SCALH EQU $538 ;HIGH BYTE OF SCALE FACTOR
05B8: 67 OFFXL EQU $5B8 ;LOW BYTE OF X-OFFSET
0638: 68 OFFXH EQU $638 ;HIGH BYTE OF X-OFFSET
06B8: 69 OFFYL EQU $6B8 ;LOW BYTE OF Y-OFFSET
0738: 70 OFFYH EQU $738 ;HIGH BYTE OF Y-OFFSET
0678: 71 HNDX EQU $678 ;TEMP INDEX FOR CURSOR PLOT
0578: 72 TEMPL EQU $578
05F8: 73 TEMPH EQU $5F8
02A5: 74 COUNT EQU $2A5 ;UTILITY COUNT REG
06F8: 75 CHAR EQU $6F8 ;TEMPORARY CHARACTER STORE
07F8: 76 MSLOT EQU $7F8 ;CURRENT SLOT POINTER %CN
0000: 77 KBD EQU $C000 ;KEYBOARD STROBE
0010: 78 KBDSTRB EQU $C010 ;KEYBOARD STROBE RESET
0050: 79 SGR EQU $C050 ;DISPLAY MODE REFERENCES
0051: 80 STEXT EQU $C051
0052: 81 SHMIX EQU $C052
0053: 82 SHIX EQU $C053
0054: 83 SPAG1 EQU $C054
0055: 84 SPAG2 EQU $C055
0056: 85 SLORES EQU $C056
0057: 86 SHIRES EQU $C057
00B1: 87 DEVO EQU $C0B1 ;BITPAD DEVICE ADDRESSES
00B0: 88 DEV1 EQU $C0B0
00B3: 89 DEV2 EQU $C0B3
00B2: 90 DEV3 EQU $C0B2
00FF: 91 ROMSW EQU $C0FF ;REFERENCE ADDRESS TO FREE %CBOO
FE93: 92 SETVID EQU $FE93 ;SET CHARACTER OUTPUT TO NORMAL
FD0D: 93 COUT EQU $FD0D ;CHARACTER OUTPUT
FF5B: 94 IORTS EQU $FF5B ;UTILITY LOCATION CONTAINING RTS
0000: 95 *****
0000: 96 *
0000: 97 *
0000: 98 * CNOO ROM ENTRY
0000: 99 * FLAG SET-UP: C CLEAR FOR INH ENTRY
0000: 100 * C SET FOR POINT RETURN (CURSOR)
0000: 101 *
0000: 102 * CNOO ENTRY CODE REPLICATED FOR EACH
0000: 103 * N FROM 9 TO F (CORRESPONDING TO 1 TO 7)
0000: 104 *
0000: 105 *****

```

```

0000: 107 *****
0000: 108 *
0000: 109 * CBOO SPACE ENTRY
0000: 110 *
0000: 111 * GET SLOT NO, SAVE MSLOT,
0000: 112 * AND SET DEFAULTS
0000: 113 *
0000: 114 *****
----- NEXT OBJECT FILE NAME IS XX
CBOO: 115 DRG %CBOO
0000: 116 DBJ %6000
CBOO: 8D A5 02 117 DTHROM STA C1HAR ;SAVE ACCUM FOR PR# ROUTINE
CBOO: 68 118 PLA ;PULL RETURN VECTOR TO
CBOA: 68 119 PLA ;GET SLOT NO.
CBO5: 28 120 PLP
CBO6: 50 01 121 DVC AOTHROM
CBO8: 60 122 RTS
CBO9: 8D FB 07 123 AOTHROM STA MSLOT ;SAVE SLOT NO.
CBOC: 8D 98 02 124 STA SAVSLOT
CBOF: AD A5 02 125 LDA C1HAR
CB12: 8D FB 06 126 STA CHAR
CB15: 48 127 PHA ;SAVE ACCUM
CB16: 8A 128 TXA ;SAVE X-REG AND Y-REG
CB17: 48 129 PHA
CB18: 98 130 TYA
CB19: 48 131 PHA
CB1A: 08 132 PHP ;SAVE STATUS
CB1B: AE FB 07 133 LDY MSLOT ;LOAD X FOR SLOT DEP VARS
CB1E: 8D 38 04 134 LDA MPAGE, X
CB21: 49 25 135 EOR %25
CB23: 5D 38 03 136 EOR PAGE, X
CB26: 29 3F 137 AND %3F
CB28: F0 02 138 BEQ PRCHK ;IF SO THEN NO DEFAULT
CB2A: 20 90 CE 139 JSR DEFAULT
CB2D: E4 37 140 PRCHK CPX COUTH ;CHECK IF FROM PR#
CB2F: D0 03 141 BNE NOPR
CB31: 4C AD CC 142 JMP SYNTAX ;IF SO, THEN TAKE IN COMMANDS
CB34: 28 143 NOPR PLP ;RECOVER STATUS
CB35: 08 144 PHP ;AND SAVE
CB36: 80 07 145 BCS EPOINT
CB38: AD FB 06 146 LDA CHAR
CB3B: A4 24 147 LDY CH
CB3D: 91 28 148 STA (BASL),Y ;ELIM FLASHING CURSOR
CB3F: 150 *****
CB3F: 151 *****
CB3F: 152 *
CB3F: 153 * MAIN LOOP ENTRY
CB3F: 154 *
CB3F: 155 *****
CB3F: 156 *****
CB3F: 20 89 CB 157 EPOINT JSR MREAD ;READ BITPAD
CB42: 2C 00 C0 158 BIT KBD
CB45: 30 26 159 BMI END
CB47: AD 90 02 160 LDA TEM ;CHECK IF PEN DOWN
CB4A: 29 03 161 AND %3
CB4C: C9 03 162 CMP %3 ;EXIT IF PEN DOWN
CB4E: D0 1D 163 BNE END
CB50: AC FB 07 164 LDY MSLOT
CB53: 99 08 03 165 LDA PAGE, Y
CB56: 29 7F 166 AND %7F
CB58: F0 0E 167 BEQ E1PNT
CB5A: 20 F0 CB 168 JSR CURSOUT ;DRAW CURSOR
CB5D: A9 60 169 LDA %60
CB5F: 20 A1 CC 170 JSR QWAIT ;LEAVE CURSOR ON FOR AWHILE
CB62: 20 F0 CB 171 JSR CURSOUT ;THEN DELETE CURSOR
CB65: AC FB 07 172 LDY MSLOT
CB68: 99 08 04 173 E1PNT LDA MPAGE, Y
CB6B: 10 D2 174 BPL EPOINT

```



```

C86D: 176 *****
C86D: 177 *****
C86D: 178 *
C86D: 179 * EXIT ROUTINES
C86D: 180 *
C86D: 181 *****
C86D: 182 *****
C86D: A2 03 183 END LDX #03
C86F: 8D 81 02 184 INEX1 LDA XFLL, X ) COPY RESULTS TO TEM REGISTERS
C872: 9D 85 02 185 STA TEMXL, X
C875: CA 186 DEX
C876: 10 F7 187 BPL INEX1
C878: AE F8 07 188 LDX HSL0T
C87D: 8D 88 03 189 LDA PAGE, X ) TEST HIGH BIT OF 'PAGE' TO
C87E: 10 03 190 BPL INEX2 ) SEE IF DATA IS TO BE SCALED
C880: 20 70 C8 191 JSR SCALE ) SCALE AND OFFSET DATA INTO TEM-A
C883: 28 192 INEX2 PLP
C884: 90 06 193 BCC INEX1T ) SKIP TO FURTHER PROCESSING
C886: 68 194 EXIT PLA
C887: A8 195 TAY
C888: 68 196 PLA
C889: AA 197 TAX
C88A: 68 198 PLA
C88B: 60 199 RTS
C88C: 200 *
C88C: 201 * SET INPUT BUFFER TO +0000, +0000, +00
C88C: 202 *
C88C: 203 *****
C88C: A9 30 204 INEX1T LDA #00
C88E: A0 0E 205 LDY #0E
C890: 99 00 02 206 GLOOP STA IN0, Y
C893: 88 207 DEY
C894: 10 FA 208 BPL GLOOP
C896: A9 AB 209 LDA #0AB
C898: 8D 00 02 210 STA #200
C89B: 8D 06 02 211 STA #206
C89E: 8D 0C 02 212 STA #20C
C8A1: A9 AC 213 LDA #0AC
C8A3: 8D 05 02 214 STA #205
C8A6: 8D 08 02 215 STA #208
C8A9: AD 80 02 216 LDA TEM
C8AC: 29 10 217 AND #10
C8AE: D0 16 218 BNE ASCEX
C8B0: AD 85 02 219 LDA TEMXL
C8B3: AC 86 02 220 LDY TEMX ) CONVERT X TO ASCII
C8B6: A2 00 221 LDX #0 ) IN INPUT BUFFER
C8B8: 20 6A CA 222 JSR ASCON
C8BB: AD 87 02 223 LDA TEMYL
C8BE: AC 88 02 224 LDY TEMY
C8C1: A2 06 225 LDX #06
C8C3: 20 6A CA 226 JSR ASCON ) CONVERT Y TO ASCII
C8C6: 2C 00 C0 227 ASCEX BIT KBD ) IN INPUT BUFFER
C8C9: 10 03 228 BPL ASC1EX
C8CB: A9 AD 229 LDA #0AD
C8CD: 8D 0C 02 230 STA #20C
C8D0: AD 80 02 231 ASC1EX LDA TEM
C8D3: 48 232 PHA
C8D4: 29 0F 233 AND #0F
C8D6: 09 80 234 ORA #00
C8D8: 8D 0E 02 235 STA #20E
C8DB: 68 236 PLA
C8DC: 4A 237 LSR A
C8DD: 4A 238 LSR A
C8DE: 4A 239 LSR A
C8DF: 4A 240 LSR A
C8E0: 29 0F 241 AND #0F
C8E2: 09 80 242 ORA #00
C8E4: 8D 0D 02 243 STA #20D
C8E7: A8 244 TAY
C8E8: 68 245 PLA
C8E9: 68 246 PLA
C8EA: 68 247 PLA
C8EB: A2 0F 248 LDX #0F
C8ED: A9 8D 249 LDA #0D
C8EF: 60 250 RTS

```

```

C8F0: 252 *****
C8F0: 253 *****
C8F0: 254 *
C8F0: 255 * CURSOR ROUTINE
C8F0: 256 *
C8F0: 257 *****
C8F0: 258 *****
C8F0: 20 70 C8 259 CURSQUT JSR SCALE
C8F3: 4C 0C C9 260 JMP C1SKP
C900: 261 ORG $100+0THROM
C900: 18 262 INPUTXY CLC
C901: 80 263 DFB $B0 ) CARRY SET FOR INH ENTRY
C902: 38 264 POINT SEC ) SKIP NEXT BYTE
C903: 88 265 CLV ) SET CARRY FOR ENTRY TO GET A POINT
C904: 08 266 PHP
C905: 78 267 BEI ) SAVE FLAGS FOR LATER
C906: 2C FF CF 268 BIT ROMSW ) DISABLE INTERRUPT UNTIL MSL0T 3F
C909: 20 00 C8 269 JSR 0THROM ) SWITCH OFF ALL $C800 ROMS
C90C: A5 2A 270 C1SKP LDA HBASL ) SWITCH TO $C800 SPACE
C90E: 48 271 PHA
C90F: A5 38 272 LDA HBASH
C911: 48 273 PHA
C912: AD 87 02 274 LDA TEMYL
C913: AE 85 02 275 LDX TEMXL
C918: 20 4C CA 276 CALLCURS JSR WINCHK
C91B: 48 277 PHA
C91C: D0 30 278 BCS OUT1
C91E: 20 F0 C9 279 JSR BASCLC
C921: 280 *****
C921: 281 *
C921: 282 * MODE EVALUATION
C921: 283 *
C921: 284 *****
C921: D9 88 03 285 LDA PAGE, Y
C924: 29 7F 286 AND #07F
C926: A8 287 TAY
C927: 29 0C 288 AND #0C
C929: F0 26 289 BEQ GR
C92B: 98 290 TYA
C92C: 29 63 291 AND #063
C92E: F0 06 292 BEQ TEXT
C930: 68 293 MIX PLA
C931: 48 294 PHA ) MIXED GRAPHICS BOUNDARY TEST
C932: C9 AD 295 CMP #160
C934: 90 18 296 BCC GR
C936: 297 *****
C936: 298 *
C936: 299 * TEXT MODE CURSOR GENERATION
C936: 300 *
C936: 301 *****
C936: 98 302 TEXT TYA
C937: 20 DE C9 303 JSR LOCCLC
C93A: D1 2A 304 LDA (HBASL), Y
C93C: 48 305 PHA
C93D: A9 DF 306 LDA #0DF
C93F: 91 2A 307 STA (HBASL), Y
C941: A9 80 308 LDA #080
C943: 20 A1 C0 309 JSR GWAIT
C946: 68 310 PLA
C947: 91 2A 311 STA (HBASL), Y
C949: A9 80 312 LDA #080
C94B: 20 A1 C0 313 JSR GWAIT
C94E: 40 D0 C9 314 OUT1 JMP OUT
C951: 98 315 GR TYA
C952: C9 0F 316 CMP #0F
C954: 80 18 317 BCS HIRES
C956: A5 2B 318 LDA HBASH
C958: 29 10 319 AND #010
C95A: 320 *****
C95A: 321 *
C95A: 322 * LD RESOLUTION GRAPHICS
C95A: 323 * CURSOR ROUTINE
C95A: 324 *
C95A: 325 *****

```



```

C95A:0B 326 LORES PHP
C95B:A9 F0 327 LDA ##F0
C95D:28 328 PLP
C95E:D0 02 329 BNE LOR1
C960:49 FF 330 EOR ##FF
C962:48 331 LOR1 PHA
C963:98 332 TVA
C964:0A 333 ASL A
C965:0A 334 ASL A
C966:20 DE C9 335 JSR LOCLC
C969:68 336 PLA
C96A:51 2A 337 EOR (HBASL),Y
C96C:91 2A 338 STA (HBASL),Y
C96E:4C D0 C9 339 JMP OUT
C971: 340 *****
C971: 341 *
C971: 342 * HIRES CURSOR ROUTINE
C971: 343 *
C971: 344 *****
C971:A9 16 345 HIRES LDA ##16
C973:8D A5 02 346 STA COUNT
C976:8A 347 TXA
C977:38 348 SEC
C978:E9 05 349 SBC ##5
C97A:AA 350 TAX
C97B:D0 03 351 DCS LOOP
C97D:CE 66 02 352 DEC TEMX
C980:A0 05 353 LDY ##5
C982:AD A5 02 354 LDA COUNT
C985:D9 D8 C9 355 CTRLDDP CMP CTRCHK,Y
C988:F0 14 356 BEQ OUTSIDE
C98A:88 357 DEY
C98B:10 FB 358 BPL CTRLDDP
C98D:68 359 PLA
C98E:20 4C CA 360 JSR WINCHK
C991:48 361 PHA
C992:80 0A 362 BCS OUTSIDE
C994:20 F0 C9 363 JSR BASCLC
C997:4C 78 D6 364 LDY HNDX
C99A:51 2A 365 EOR (HBASL),Y
C99C:91 2A 366 STA (HBASL),Y
C99E:A9 0C 367 OUTSIDE LDA ##C
C9A0:CD A5 02 368 CMP COUNT
C9A3:F0 0E 369 BEQ A
C9A5:80 1D 370 BCS B
C9A7:E8 371 INX
C9A8:D0 03 372 BNE C
C9AA:EE 86 02 373 INC TEMX
C9AD:CE A5 02 374 C DEC COUNT
C9B0:4C 80 C9 375 JMP LOOP
C9B3:8A 376 A TXA
C9B4:E9 05 377 SBC ##5
C9B6:AA 378 TAX
C9B7:AD 86 02 379 LDA TEMX
C9BA:E9 00 380 SBC ##0
C9BC:8D 86 02 381 STA TEMX
C9BF:68 382 PLA
C9C0:38 383 SEC
C9C1:E9 06 384 SBC ##6
C9C3:46 385 PHA
C9C4:68 386 B PLA
C9C5:18 387 CLC
C9C6:69 01 388 ADC ##1
C9C8:48 389 PHA
C9C9:CE A5 02 390 DEC COUNT
C9CC:F0 02 391 BEQ OUT
C9CE:D0 80 392 BNE LOOP
C9D0:68 393 OUT PLA
C9D1:68 394 PLA
C9D2:85 2B 395 STA HBASH
C9D4:68 396 PLA
C9D5:85 2A 397 STA HBASL
C9D7:60 398 RTS

```

```

C9DB:12 399 CTRCHK DFB #12
C9DD:11 400 DFB #11
C9DA:10 401 DFB #10
C9DB:07 402 DFB #7
C9DC:06 403 DFB #6
C9DD:05 404 DFB #5
C9DE: 405 *****
C9DE: 406 * TEXT BASE CALC
C9DE: 407 *
C9DE: 408 * THIS SUBROUTINE SETS UP BASE REGISTER
C9DE: 409 * FOR LORES OR TEXT
C9DE: 410 * ENTER WITH 'PAGE' IN A --
C9DE: 411 * EXIT WITH HNDX IN Y, READY FOR SCREEN PROCESSING
C9DE: 412 *
C9DE: 413 *****
C9DE:48 414 LOCLC PHA
C9DF:A5 2B 415 LDA HBASH
C9E1:29 03 416 AND #3
C9E3:85 2B 417 STA HBASH
C9E5:68 418 PLA
C9E6:29 0C 419 AND ##0C
C9E8:05 2B 420 ORA HBASH
C9EA:85 2B 421 STA HBASH
C9EC:AC 78 D6 422 LDY HNDX
C9EF:60 423 RTS
C9F0: 425 *****
C9F0: 426 *
C9F0: 427 * SCREEN BASE ADDRESS CALC
C9F0: 428 *
C9F0: 429 * ENTER WITH LO BYTE OF Y IN ACCUM
C9F0: 430 * AND WITH LO BYTE OF X IN X-REG
C9F0: 431 *
C9F0: 432 * BASE ADDRESS WILL BE COMPUTED INTO
C9F0: 433 * HBASL,HBASH, AND HNDX: HIRES BIT ADDRESSED
C9F0: 434 * IS RETURNED AS A 1 IN A BIT IN THE ACCUM
C9F0: 435 *
C9F0: 436 *****
C9F0:48 437 BASCLC PHA
C9F1:29 C0 438 AND ##C0
C9F3:85 2A 439 STA HBASL
C9F5:4A 440 LSR A
C9F6:4A 441 LSR A
C9F7:05 2A 442 ORA HBASL
C9F9:85 2A 443 STA HBASL
C9FB:68 444 PLA
C9FC:4C 0C CA 445 JMP C2SKP
CA00: 446 ORG $200+OTHRON
CA00:18 447 CLC
CA01:80 448 DFB #B0
CA02:38 449 SEC
CA03:88 450 CLV
CA04:08 451 PHP
CA05:78 452 SEI
CA06:2C FF CF 453 BIT ROMSW
CA09:20 00 C8 454 JSR OTHRON
CA0C:85 2B 455 C2SKP STA HBASH
CA0E:0A 456 ASL A
CA0F:0A 457 ASL A
CA10:0A 458 ASL A
CA11:26 2B 459 ROL HBASH
CA13:0A 460 ASL A
CA14:26 2B 461 ROL HBASH
CA16:0A 462 ASL A
CA17:66 2A 463 ROR HBASL
CA19:A5 2B 464 LDA HBASH
CA1B:29 1F 465 AND ##1F
CA1D:85 2B 466 STA HBASH
CA1F:8A 467 TXA
CA20:AC 86 02 468 LDY TEMX
CA23:C0 00 469 CPY ##0
CA25:F0 05 470 BEQ HPOSN2
CA27:A0 23 471 LDY ##23
CA29:69 04 472 ADC ##4

```

; \$C200 SPACE ENTRY


```

CA2B:08      473 HPOSN1    INV
CA2C:E9 07    474 HPOSN2    SBC    #17
CA2E:30 FB    475          BCS    HPOSN1
CA30:69 08    476          ADC    #18
CA32:BC 7B 06 477          STY    HNDX
CA35:AB       478          TAY
CA36:A9 00    479          LDA    #10
CA3B:2B       480          SEC
CA39:2A       481 CLOOP     ROL    A
CA3A:BB       482          DEY
CA3B:D0 FC    483          BNE    CLOOP
CA3D:AC FB 07 484          LDY    HSL0T
CA40:4B       485          PHA
CA41:29 BB 03 486          LDA    PAGE, Y
CA44:29 60    487          AND    #160
CA46:05 2B    488          ORA    HBASH
CA4B:05 2B    489          STA    HBASH
CA4A:6B       490          PLA
CA4B:60       491          RTS
CA4C:         492          CHN    BITPAD35, 2P
CA4C:         3 *****
CA4C:         4 *
CA4C:         5 * WINDOW CHECK SUBROUTINE
CA4C:         6 *
CA4C:         7 * ENTER WITH LO BYTE OF Y IN ACCUM
CA4C:         8 * AND LO BYTE OF X IN X-REG
CA4C:         9 *
CA4C:        10 * RETURN WILL BE CARRY CLEAR IF
CA4C:        11 * WITHIN WINDOW AND CARRY SET IF
CA4C:        12 * OUTSIDE
CA4C:        13 *****
CA4C:4B       14 WINCHK     PHA
CA4D:C9 00    15          CMP    #192
CA4F:30 16    16          BCS    NO
CA51:AD 86 02 17          LDA    TEMX
CA54:F0 09    18          BEQ    YES
CA56:C9 02    19          CMP    #12
CA5B:30 00    20          BCS    NO
CA5A:8A       21          TXA
CA5B:C9 18    22          CMP    #24
CA5D:30 08    23          BCS    NO
CA5F:AD 88 02 24 YES      LDA    TEMY
CA62:D0 03    25          BNE    NO
CA64:1B       26          CLC
CA65:6B       27          PLA
CA66:60       28          RTS
CA67:3B       29 NO       SEC
CA6B:6B       30          PLA
CA69:60       31          RTS
CA6A:         32 *****
CA6A:         33 *
CA6A:         34 *
CA6A:         35 * ASCII CONVERSION
CA6A:         36 * ENTER WITH HIGH BYTE
CA6A:         37 * IN Y-REG. LOW BYTE IN ACCUM
CA6A:         38 * AND INPUT BUFFER OFFSET
CA6A:         39 * IN X-REG
CA6A:         40 *
CA6A:         41 *****
CA6A:4B       42 ASCON      PHA
CA6B:9B       43          TYA
CA6C:10 12    44          BPL    POSIT
CA6E:4B       45          PHA
CA6F:A9 AD    46          LDA    #1AD
CA71:9D 00 02 47          STA    INQ, X
CA74:6B       48          PLA
CA75:66 2A    49          STX    HBASL
CA77:AA       50          TAX
CA7B:6B       51          PLA
CA79:20 5B 0B 52          JSR    TWOCDM
CA7C:4B       53          PHA
CA7D:BA       54          TIA
CA7E:A6 2A    55          LDY    HBASL

```

```

CAB0:AB       56 POSIT     TAY
CAB1:EB       57          INX
CAB2:6B       58          PLA
CAB3:3B       59          SEC
CAB4:30 03    60          BCS    ASKIP
CAB6:FE 00 02 61 FLOOP     INC    INQ, X
CAB9:E9 E8    62 ASKIP     SBC    #1EB
CABB:4B       63          PHA
CABD:9B       64          TYA
CABD:E9 03    65          SBC    #13
CABF:AB       66          TAY
CA90:6B       67          PLA
CA91:30 F3    68          BCS    FLOOP
CA93:69 E8    69          ADC    #1EB
CA95:4B       70          PHA
CA96:9B       71          TYA
CA97:69 03    72          ADC    #13
CA99:AB       73          TAY
CA9A:6B       74          PLA
CA9B:30 03    75          BCS    BSKIP
CA9D:FE 01 02 76 HLOOP     INC    IN1, X
CAA0:3B       77 BSKIP     SEC
CAA1:E9 64    78          SBC    #164
CAA3:30 FB    79          BCS    HLOOP
CAA5:8B       80          DEY
CAA6:10 F5    81          BPL    HLOOP
CAAB:1B       82          CLC
CAA9:C8       83          INY
CAAA:69 64    84          ADC    #164
CAAC:3B       85          SEC
CAAD:30 03    86          BCS    CSKIP
CAAF:FE 02 02 87 JLOOP     INC    IN2, X
CAB2:E9 0A    88 CSKIP     SBC    #1A
CAB4:30 F9    89          BCS    JLOOP
CAB6:69 0A    90          ADC    #1A
CAB8:30 03    91          BCS    DSKIP
CABA:FE 03 02 92 KLOOP     INC    INQ, X
CABD:E9 01    93 DSKIP     SBC    #11
CABF:30 F9    94          BCS    KLOOP
CAC1:60       95 RTN       RTS
CAC2:         97 *****
CAC2:         98 *
CAC2:         99 * OFFSET AND DIVIDE ROUTINE
CAC2:        100 *
CAC2:        101 * ENTER WITH OFFSET IN REGH
CAC2:        102 * AND REGL. VALUE TO SCALE IN
CAC2:        103 * I (HIGH BYTE) AND A (LOW BYTE)
CAC2:        104 * SCALE (INTEGER DIVISOR) IN
CAC2:        105 * DIVH AND DIVL
CAC2:        106 *
CAC2:        107 * RESULT IN REGH AND REGL
CAC2:        108 * VALUE MAY BE +OR-32767
CAC2:        109 * SCALE 0 TO +32767
CAC2:        110 * RESULT IS TWO'S COMPLEMENT
CAC2:        111 * REMAINDER IS LOST
CAC2:        112 *
CAC2:        113 *****
CAC2:AC FB 07 114 OFFDIV     LDY    HSL0T
CAC5:4B       115          PHA
CAC6:89 3B 04 116          LDA    MPAGE, Y
CAC9:0A       117          ASL    A
CACA:30 05    118          BMT    DIVIDE
CACC:6B       119          PLA
CACE:20 64 0B 120          JSR    OFFSET
CADD:4B       121          PHA
CAD1:BB       122 DIVIDE     CLV
CAD2:A0 00    123          LDY    #10
CAD4:AD A4 02 124          LDA    DIVH
CAD7:C8       125 LOOP1     INY
CADB:0E A3 02 126          ASL    DIVL
CADB:2A       127          ROL    A
CADC:10 F9    128          BPL    LOOP1
CADE:BD A4 02 129          STA    DIVH

```


CAE1: A9 00	130	LDA	#0
CAE3: BD 88 02	131	STA	REGH
CAE6: BD 87 02	132	STA	REGL
CAE9: 68	133	PLA	
CAEA: E8	134	INX	
CAEB: CA	135	DEX	
CAEC: 08	136	PHP	
CAED: 10 03	137	BPL	POS
CAEF: 20 58 C9	138	JSR	TWOCOM
CAF2: 08	139	SEC	
CAF3: ED A3 02	140	SBC	DIVL
CAF6: 48	141	PHA	
CAF7: 8A	142	TXA	
CAF8: ED A4 02	143	SBC	DIVH
CAF9: AA	144	TAX	
CAFC: 68	145	PLA	
CAFD: 4C 0C C8	146	JMP	LOOP3
CB00:	147	ORG	\$300+DTHROM
CB00: 18	148	CLC	
CB01: 80	149	DFB	\$80
CB02: 08	150	SEC	
CB03: 88	151	CLV	
CB04: 08	152	PHP	
CB05: 78	153	SEI	
CB06: 2C FF CF	154	BIT	ROMSW
CB09: 20 00 C8	155	JSR	OTHROM
CB0C: 08	156	PHP	
CB0D: 2E 87 02	157	RCL	REGL
CB10: 2E 88 02	158	RCL	REGH
CB13: 88	159	DEY	
CB14: 30 16	160	BMI	FEXIT
CB16: 4E A4 02	161	LSR	DIVH
CB19: 6E A3 02	162	ROR	DIVL
CB1C: 28	163	PLP	
CB1D: 80 D4	164	BCS	LOOP2
CB1F: 6D A3 02	165	ADC	DIVL
CB22: 48	166	PHA	
CB23: 8A	167	TXA	
CB24: 6D A4 02	168	ADC	DIVH
CB27: AA	169	TAX	
CB28: 68	170	PLA	
CB29: 4C 0C C8	171	JMP	LOOP3
CB2C: 28	172	PLP	
CB2D: 28	173	PLP	
CB2E: 10 0F	174	BPL	EEXIT
CB30: AD 87 02	175	LDA	REGL
CB33: AE 88 02	176	LDX	REGH
CB36: 20 58 C8	177	JSR	TWOCOM
CB39: 8E 88 02	178	STX	REGH
CB3C: BD 87 02	179	STA	REGL
CB3F: AC F8 07	180	LDY	MSLOT
CB42: B9 38 04	181	LDA	MPAGE.Y
CB45: 0A	182	ASL	A
CB46: 10 0F	183	BPL	EEXIT
CB48: AD 87 02	184	LDA	REGL
CB4B: AE 88 02	185	LDY	REGH
CB4E: 20 64 C8	186	JSR	OFFSET
CB51: BD 87 02	187	STA	REGL
CB54: BE 88 02	188	STX	REGH
CB57: 60	189	RTS	
CB58: 49 FF	190	EOR	#FF
CB5A: 18	191	CLC	
CB5B: 69 01	192	ADC	#1
CB5D: 48	193	PHA	
CB5E: 8A	194	TXA	
CB5F: 49 FF	195	EOR	#FF
CB61: AA	196	TAX	
CB62: 68	197	PLA	
CB63: 60	198	RTS	
CB64: 38	199	SEC	
CB65: ED A1 02	200	SBC	DREGL
CB68: 48	201	PHA	
CB69: 8A	202	TXA	

CB6A: ED A2 02	203	SBC	DREGL
CB6D: AA	204	TAX	
CB6E: 68	205	PLA	
CB6F: 60	206	RTS	
CB70:	208	*****	
CB70:	209	*	
CB70:	210	* SCALE ROUTINE	
CB70:	211	*	
CB70:	212	* VALUES IN -FL- REGISTERS CONVERTED	
CB70:	213	* TO SCALED VALUES IN TEM- REGISTERS	
CB70:	214	*	
CB70:	215	*****	
CB70: AC F8 07	216	SCALE	LDY MSLOT
CB73: B9 88 04	217		LDA SCALL.Y
CB76: BD A3 02	218		STA DIVL
CB79: B9 38 05	219		LDA SCALH.Y
CB7C: BD A4 02	220		STA DIVH
CB7F: B9 88 05	221		LDA OFFXL.Y
CB82: BD A1 02	222		STA DREGL
CB85: B9 38 06	223		LDA OFFXH.Y
CB88: BD A2 02	224		STA DREGL
CB8B: AD 81 02	225		LDA XFLL
CB8E: AE 82 02	226		LDX XFLH
CB91: 20 C2 CA	227		JSR OFFDIV
CB94: AC F8 07	228		LDY MSLOT
CB97: AD 87 02	229		LDA TEMYL
CB9A: BD 85 02	230		STA TEMXL
CB9D: AD 88 02	231		LDA TEMY
CB90: BD 86 02	232		STA TEMX
CB93: B9 88 06	233		LDA OFFYL.Y
CB96: BD A1 02	234		STA DREGL
CB99: B9 38 07	235		LDA OFFYH.Y
CB9C: BD A2 02	236		STA DREGL
CB9F: AD 83 02	237		LDA YFLL
CB82: AE 84 02	238		LDX YFLH
CB85: 20 C2 CA	239		JSR OFFDIV
CB88: 60	240		RTS
CB89:	242	*****	
CB89:	243	*	
CB89:	244	*	
CB89:	245	* TRIPLE READ OF BITPAD	
CB89:	246	* IF OFFSCALE THEN A 1 IS	
CB89:	247	* PUT IN THE HIGH NIBBLE OF	
CB89:	248	* TEM	
CB89:	249	*	
CB89:	250	*	
CB89:	251	*****	
CB89: 20 F4 C8	252	MREAD	JSR MREAD
CB8C: 90 12	253	BCC	SWCHK
CB8E: 2C 00 C0	254		BIT KBD
CB81: 30 03	255		BMI OFFB1
CB83: 4C B9 C8	256		JMP MREAD
CB86: 48	257	OFFB1	PHA
CB87: AD 80 02	258		LDA TEM
CB8A: 09 08	259		ORA #08
CB8C: BD 80 02	260		STA TEM
CB8F: 68	261		PLA
CB90: 48	262	SWCHK	PHA
CB91: 4D 80 02	263		EOR TEM
CB94: 6A	264		ROR A
CB95: 90 05	265		BCC NOSWITCH
CB97: A7 50	266		LDA #50
CB99: 20 A1 CC	267		JSR WAIT
CB9C: 68	268	NOSWITCH	PLA
CB9D: 6A	269		ROR A
CB9E: AD 80 02	270		LDA TEM
CB91: 2A	271		RCL A
CB92: 29 13	272		AND #13
CB94: BD 80 02	273		STA TEM
CB97: 4E 84 02	274		LSR YFLH
CB9A: 6E 83 02	275		ROR YFLL
CB9D: 4E 82 02	276		LSR XFLH
CB9F: 6E 81 02	277		ROR XFLL


```

CBF3: 60      278      RTS
CBF4: AD F8 07 279 MIREAD LDA MSL0T
CBF7: 0A      280      ASL A
CBF8: 0A      281      ASL A
CBF9: 0A      282      ASL A
CBFA: 0A      283      ASL A
CBFB: AA      284      TAX
CBFC: 40 03    285      LDY #53
CBFE: D0 0C    286      BNE C4SKP
CC00:      287      ORG $400+0THROM
CC00: 18      288      CLC
CC01: 80      289      DFB #B0
CC02: 38      290      SEC
CC03: 88      291      CLV
CC04: 08      292      PHP
CC05: 78      293      SEI
CC06: 2C FF CF 294      BIT R0MSW
CC09: 20 00 C8 295      JSR 0THROM
CC0C: A9 00    296      LDA #50
CC0E: 99 81 02 297 ZDLOOP STA XFLH, Y
CC11: 88      298      DEY
CC12: 10 FA    299      BPL ZDLOOP
CC14: A0 06    300      LDY #56
CC16: 8C 78 06 301      STY HNDX
CC19: 20 83 CC 302 RDLOOP JSR RESLP
CC1C: 80 81 00 303      LDA DEV0, X
CC1F: 20 42 CC 304      JSR READTAB
CC22: 80 1D    305      BCS OFFSC
CC24: A0 02    306      LDY #52
CC26: 20 8E CC 307      JSR AMOVE
CC29: 20 83 CC 308      JSR RESLP
CC2C: BD 80 C0 309      LDA DEV1, X
CC2F: 20 42 CC 310      JSR READTAB
CC32: 80 0D    311      BCS OFFSC
CC34: A8      312      PHA
CC35: A0 00    313      LDY #50
CC37: 20 6E CC 314      JSR AMOVE
CC3A: 68      315      PLA
CC3B: CE 78 06 316      DEC HNDX
CC3E: D0 09    317      BNE RDLOOP
CC40: 18      318      CLC
CC41: 60      319 OFFSC RTS
CC42: A9 12    320 READTAB LDA #512
CC44: E9 01    321 ALDOP SBC #51
CC46: D0 FC    322      DNE ALDOP
CC48: BD 82 C0 323      LDA DEV3, X
CC4B: 0A      324      ASL A
CC4C: 0A      325      ASL A
CC4D: 0A      326      ASL A
CC4E: 0A      327      ASL A
CC4F: 49 70    328      EOR #570
CC51: 29 F0    329      AND #5F0
CC53: BD B7 02 330      STA TEMYL
CC56: BD B3 C0 331      LDA DEV2, X
CC59: BD B8 02 332      STA TEMY
CC5C: A9 60    333      LDA #540
CC5E: E9 01    334 AILLOOP SBC #51
CC60: D0 FC    335      DNE AILLOOP
CC62: A0 04    336      LDY #54
CC64: 4E 88 02 337 DLOOP LSR TEMY
CC67: 6E 87 02 338      ROR TEMYL
CC6A: 68      339      DEY
CC6B: D0 F7    340      DNE DLOOP
CC6D: AD 89 02 341      LDA TEMY
CC70: D0 07    342      BNE ATST
CC72: A9 60    343      LDA #560
CC74: CD B7 02 344      CMP TEMYL
CC77: 90 02    345      BCC BTST
CC79: C9 0A    346 ATST CMP #50A
CC7B: BD B2 C0 347 BTST LDA DEV3, X
CC7E: A9 01    348      EOR #51
CC80: 29 01    349      AND #51
CC82: 60      350      RTS

```

```

CC83: AD FB 07 351 RESLP LDA MSL0T
CC86: 48      352      PHA
CC87: A9 03    353      LDA #53
CC89: 48      354      PHA
CC8A: 2C 41 CC 355      BIT OFFSC
CC8D: 60      356      RTS
CC8E: AD B7 02 357 AMOVE LDA TEMYL
CC91: 79 B1 02 358      ADC XFLH, Y
CC94: 99 B1 02 359      STA XFLH, Y
CC97: AD B8 02 360      LDA TEMY
CC9A: 79 B2 02 361      ADC XFLH, Y
CC9D: 99 B2 02 362      STA XFLH, Y
CCA0: 60      363      RTS
CCA1: 28      364 GWAIT SEC
CCA2: 48      365 G2WAIT PHA
CCA3: E9 01    366 G3WAIT SBC #51
CCA5: D0 FC    367      BNE G3WAIT
CCA7: 68      368      PLA
CCAB: E9 01    369      SBC #51
CCAA: D0 F6    370      BNE G2WAIT
CCAC: 60      371      RTS
CCAD:      372 *****
CCAD:      373 *****
CCAD:      374 *
CCAD:      375 * CHARACTER ENTRY SEQUENCE
CCAD:      376 *
CCAD:      377 * CHARACTER STRING ANALYSIS
CCAD:      378 * S... NNNN SET SCALE FACTOR TO NNNN
CCAD:      379 * X... NNNN SET X OFFSET TO NNNN
CCAD:      380 * Y... NNNN SET Y OFFSET TO NNNN
CCAD:      381 * M... (1 OR 2) SET MIXED HIRES MODE
CCAD:      382 * H... (1 OR 2) SET HIRES MODE
CCAD:      383 * Q... (1 OR 2) SET MIXED LORES MODE
CCAD:      384 * L... (1 OR 2) SET LORES MODE
CCAD:      385 * T... (1 OR 2) SET TEXT MODE
CCAD:      386 * N... SUPPRESS ALL PRINTING
CCAD:      387 * D... RESTORE DEFAULT PARAMETERS
CCAD:      388 * C... NO CURSOR
CCAD:      389 * F... DATA RETURNED UNSCALED
CCAD:      390 * R... DATA RETURNED SCALED
CCAD:      391 * P... STREAM MODE ON
CCAD:      392 * G... STREAM MODE OFF (DEFAULT)
CCAD:      393 * A... OFFSET AFTER SCALE
CCAD:      394 * B... OFFSET BEFORE SCALE (DEFAULT)
CCAD:      395 *
CCAD:      396 *****
CCAD:      397 SYNTAX PLP
CCAE: 80 0C    398      DCS SYNT1
CCB0: A9 00    399      LDA #50
CCB2: BD 98 02 400      STA INX
CCB5: BD 99 02 401      STA NFLAG
CCB8: A9 02    402      LDA #52
CCBA: B5 36    403      STA COUTL
CCBC: AD FB 06 404 SYNT1 LDA CHAR
CCBF: C9 AD    405      CMP #5AD
CCC1: D0 03    406      BNE SYNT2
CCC3: BD A0 02 407      STA MIFLAG
CCC6: C9 A0    408 SYNT2 CMP #5A0
CCC8: F0 29    409      BEQ EXIT4
CCCA: C9 AC    410      CMP #5AC
CCCC: F0 28    411      BEQ PROC1
CCCE: C9 8D    412      CMP #5BD
CCD0: F0 26    413      BEQ PROC2
CCD2: 08      414      PHP
CCD3: AE 98 02 415      LDX INX
CCD6: F0 0C    416      BEQ CHOUT
CCD8: A9 80    417      EOR #580
CCDA: C9 0A    418      CMP #5A
CCDC: 90 06    419      BCC CHOUT
CCDE: E0 01    420      CPY #51
CCDF: F0 10    421      BEQ EXIT3
CCE0: D0 6C    422      DNE ERR1
CCE4: E0 06    423 CHOUT CPY #56
CCE6: 80 68    424      DCS ERR1

```


103


```

CE0E: BD A0 02 571 STA HIFLAG
CE11: BD 98 02 572 STA INX
CE14: AC F8 07 573 LDY MSLDT
CE17: B9 38 04 574 LDA MPAGE, Y
CE1A: 29 C0 575 AND #1C0
CE1C: 99 38 04 576 STA MPAGE, Y
CE1F: B9 88 03 577 LDA PAGE, Y
CE22: 29 3F 578 AND #3F
CE24: 49 25 579 EOR #25
CE26: 19 38 04 580 ORA MPAGE, Y
CE29: 99 38 04 581 STA MPAGE, Y
CE2C: 28 582 PLP
CE2D: 80 12 583 BCS EXIT2
CE2F: AD 99 02 584 LDA NFLAG
CE32: F0 0A 585 BEQ PRINT
CE34: A9 58 586 LDA #IORTS
CE36: B5 36 587 STA COUTL
CE38: A9 FF 588 LDA #IORTS/256
CE3A: 85 37 589 STA COUTH
CE3C: D0 03 590 DNE EXIT2
CE3E: 20 93 FE 591 PRINT JSR SETVID
CE41: 4C 86 C8 592 EXIT2 JMP EXIT
CE44: 8D 593 STRIN DFB #BD
CE45: D2 CF 02 594 ASC "RORRE XATHYS TELBAT ***"
CE48: D2 C5 A0
CE4B: D8 C1 D4
CE4E: CE D9 D3
CE51: A0 04 C5
CE54: CC C2 C1
CE57: D4 A0 AA
CE5A: AA AA 87
CE5D: D0 D1 C1 595 TABL ASC "FGABSSXYTMHLGCFRND"
CE60: C2 D3 D8
CE63: D9 D4 CD
CE66: C8 CC C7
CE69: C3 C6 D2
CE6C: CE C4
CE6E: E9 596 ADR DFB CROUT-1
CE6F: E9 597 DFB CROUT-1
CE70: E9 598 DFB CROUT-1
CE71: E9 599 DFB CROUT-1
CE72: 88 600 DFB BROUT-1
CE73: 88 601 DFB BROUT-1
CE74: 88 602 DFB BROUT-1
CE75: 86 603 DFB AROUT-1
CE76: 86 604 DFB AROUT-1
CE77: 86 605 DFB AROUT-1
CE78: 86 606 DFB AROUT-1
CE79: 86 607 DFB AROUT-1
CE7A: 23 608 DFB ROUTIN-1
CE7B: 23 609 DFB ROUTIN-1
CE7C: 23 610 DFB ROUTIN-1
CE7D: 23 611 DFB ROUTIN-1
CE7E: 23 612 DFB ROUTIN-1
CE7F: 80 613 PARAM DFB #80
CE80: 7F 614 DFB #7F
CE81: 40 615 DFB #40
CE82: 3F 616 DFB #3F
CE83: 04 617 DFB #04
CE84: 05 618 DFB #05
CE85: 06 619 DFB #06
CE86: 08 620 DFB #08
CE87: 48 621 DFB #48
CE88: 40 622 DFB #40
CE89: 02 623 DFB #02
CE8A: 0A 624 DFB #0A
CE8B: 00 625 DFB #0
CE8C: 01 626 DFB #01
CE8D: 80 627 DFB #80
CE8E: CE 628 DFB #CE
CE8F: C4 629 DFB #C4

```

```

CE90: 631 *****
CE90: 632 *****
CE90: 633 *
CE90: 634 * DEFAULT
CE90: 635 *
CE90: 636 * SCALE = 16
CE90: 637 * X OFFSET = 1536
CE90: 638 * Y OFFSET = 1536
CE90: 639 * HIRES PAGE 2
CE90: 640 * PRINT ON
CE90: 641 * CURSOR ON
CE90: 642 * STREAM MODE OFF
CE90: 643 * DATA RETURNED UNSCALED
CE90: 644 * OFFSET BEFORE SCALE
CE90: 645 *
CE90: 646 *****
CE90: 647 *****
CE90: A9 10 648 DEFAULT LDA #510
CE92: 9D 88 04 649 STA SCALL, X
CE95: A9 00 650 LDA #50
CE97: 9D 38 05 651 STA SCALL, X ;SET SCALE FOR 16
CE9A: 9D 88 05 652 STA OFFXL, X
CE9D: 9D 88 06 653 STA OFFYL, X ;SET X OFFSET TO 1536
CEA0: 8D 98 02 654 STA INX
CEA3: A9 03 655 LDA #3
CEA5: BD 80 02 656 STA TEM
CEA8: A9 06 657 LDA #6 ;SET Y OFFSET TO 1536
CEAA: 9D 38 06 658 STA OFFXH, X
CEAD: 9D 38 07 659 STA OFFYH, X
CEB0: 2C 10 C0 660 BIT KBDSTRB
CEB3: 20 F4 CD 661 JSR MIREAD
CEB6: 20 D0 CD 662 JSR SWCHK
CEB9: AE F8 07 663 LDX MSLDT
CEBC: A9 40 664 LDA #40 ;SET PAGE FOR HIRES P2
CEBE: AB 665 STMODE TAY
CEBF: 8D 52 C0 666 STA SMIX
CEC2: 8D 57 C0 667 STA SHIRES
CEC5: 8D 51 C0 668 STA STXT
CEC8: 8D 54 C0 669 STA SPAG1
CECB: 29 0C 670 AND #50C
CECD: F0 03 671 BEQ DEF1
CECF: 8D 53 C0 672 STA SMIX
CED2: 98 673 DEF1 TYA
CED3: 29 63 674 AND #63
CED5: F0 03 675 BEQ DEF2
CED7: AD 50 C0 676 LDA SGR
CEDA: 98 677 DEF2 TYA
CEDB: 29 4A 678 AND #4A
CEDD: F0 03 679 BEQ DEF3
CEDF: AD 55 C0 680 LDA SPAG2
CEE2: 98 681 DEF3 TYA
CEE3: 29 03 682 AND #503
CEE5: F0 03 683 BEQ DEF4
CEE7: 8D 56 C0 684 STA SLORES
CEEA: 98 685 DEF4 TYA
CEEB: 9D 88 03 686 STA PAGE, X ;AND FOR FULL SCALE OUTPUT
CEEC: 29 3F 687 AND #3F
CEED: 49 25 688 EOR #25
CEE2: 9D 38 04 689 STA MPAGE, X
CEFD: 60 690 RTS
CEFE: 691 *****
CEFE: 692 *
CEFE: 693 * JUMP TABLE
CEFE: 694 *
CEFE: 695 *****
CEFE: 696 DRG #6F6+OTHRDM
CEFE: 4C 4C CA 697 JWINCHA JMP WINCHA
CEFE: 4C B9 C8 698 JMREAD JMP MREAD
CEFE: 4C 70 C8 699 JSSCALE JMP SCALE
*** SUCCESSFUL ASSEMBLY: NO ERRORS

```


C08E A1LOOP
 C809 A0THROM
 C8D0 ASC1EX
 C079 ATST
 C089 BROUT
 C90C C1SKP
 C091B CALLCURS
 CA39 CLOOP
 36 COUTL
 C9AD C
 C085 CTRLLOOP
 CEEA DEF4
 C083 DEV2
 C0A2 DIG2
 C0A4 DIVH
 C0BD DSXIP
 C055 EMSG1
 C050 ERR1
 C04C EXIT5
 C02C EXIT1
 C090 GLOOP
 2A HBASL
 CA2B HPOSN1
 C022 IN2
 C0B3 INEX2
 FF5B IORTS
 C0EF6 JWINCHK
 C9DE LOCLC
 C00C L00P3
 CBF4 M1READ
 C089 MREAD
 C048 NOCR6
 CAC2 OFFDIV
 C038 OFFXH
 C0A2 OREGH
 C99E OUTSIDE
 C0902 POINT
 C03E PRINT
 C042 READTAB
 CFFF ROMSW
 C0AC1 RTN
 C48B SCALL
 C056 SLORES
 C055 SPAG2
 C3D0 SWCHK
 C05D TABL
 C086 TEMX
 C936 TEXT
 C021 XFLL
 C00E ZDLOOP
 24 CH
 36 COUTL
 C022 IN2
 C022 XFLH
 C086 TEMX
 C089 REGH
 C098 SAVSLOT
 C0A3 DIVL
 C088 PAGE
 C079 TEMPL
 C051 TEXT
 C055 SPAG2
 C081 DEV0
 C809 A0THROM
 C868 E1PNT
 C886 EXIT
 C8D0 ASC1EX
 C90C C1SKP
 C94E OUT1
 C971 HIRES
 C9AD C

CE6E ADR
 CD67 AROUT
 C8C6 ASCEX
 C9C4 B
 CAA0 BSKIP
 CA0C C2SKP
 CCE4 CHOUT
 C0A5 COUNT
 C0F7 CROUT2
 CDEA CROUT
 CED2 DEF1
 CE90 DEFAULT
 C082 DEV3
 CDCB DIG3
 CAD1 DIVIDE
 CB57 E1EXIT
 C053 ENSQ
 CD4F ERR3
 C886 EXIT
 C086 FLOOP
 C951 GR
 C971 HIRES
 CA2C HPOSN2
 C023 IN3
 C0B3 INEXIT
 CAAF JLOOP
 C000 KBD
 C980 LOOP
 C0B1F LOOP4
 C0A0 MIFLAG
 C07B MSL0T
 C034 NOPR
 C8C6 OFFS1
 C5B8 OFFXL
 C0A1 OREGL
 C900 OUT
 CAB0 POSIT
 CCF6 PROC1
 C088 REGH
 CD31 ROUT1
 C02B SAVSLOT
 FE93 SETVID
 C053 SMIX
 C051 STEXT
 C03C SYNT1
 C0F8 TEMPH
 C027 TEMYL
 C858 TWOCOM
 CASF YES
 2B BASL
 37 COUTH
 C023 IN3
 C023 YFLL
 C027 TEMYL
 C090 INA
 C0A0 MIFLAG
 C0A4 DIVH
 C43B MPAGE
 C5B8 OFFXL
 C000 KBD
 C052 SMIX
 C056 SLORES
 C082 DEV3
 C82D PRCHK
 C86D END
 C88C INEXIT
 C8F0 CURSOUT
 C091B CALLCURS
 C951 GR
 C980 L00P
 C983 A

CD0E ALPH1
 C9B3 A
 CA6A ASCON
 C9F0 BASCLC
 CC7B BTST
 CC0C C4SKP
 24 CH
 FDED COUT
 CDFA CROUT3
 CAB2 CSKIP
 CEDA DEF2
 C081 DEV0
 C082D DEXIT
 CDD9 DIG4
 C0A3 DIVL
 C868 E1PNT
 C86D END
 CE0C EXIT1
 CE41 EXIT2
 CCA2 G2WAIT
 CCA1 GWAIT
 CA9D HLOOP
 C020 IN0
 C090 INA
 C0900 INPUTXY
 C0EF9 JHREAD
 C010 K3DSTRB
 CAD7 LOOP1
 C962 LOR1
 C0930 MIX
 C099 NFLAG
 CD7C NOROL
 CC41 OFFSC
 C073 OFFYH
 C800 OTHROM
 C3B8 PAGE
 CAF2 POS
 CCFB PROC2
 C087 REGL
 CD3D ROUT2
 C870 SCALE
 C050 SGR
 C052 SNMIX
 CE8E STMODE
 C0C6 SYNT2
 C078 TEMPL
 C088 TEMY
 CA4C WINCHK
 C084 YFLH
 2A HBASL
 C020 IN0
 C020 TEM
 C024 YFLH
 C027 REGL
 C098 INA
 C0A1 OREGL
 C0A3 C1HAR
 C48B SCALL
 C0F8 TEMPH
 C06B CHAR
 C010 K3DSTR3
 C053 SMIX
 C057 SHIRES
 C083 DEV2
 C834 NOPR
 C86F INEX1
 C890 GLOOP
 C0900 INPUTXY
 C0930 MIX
 C095A LORES
 C985 CTRL00P
 C9C4 B

CC8E AMOVE
 CC44 ALOOP
 CAB9 ASKIP
 2B BASL
 C0A3 C1HAR
 CD0C C5SKP
 C0F8 CHAR
 37 COUTH
 C8F0 CURSOUT
 C908 CTRCHK
 CEE2 DEF3
 C080 DEV1
 C088 DIG1
 CD90 DIGLP
 CC64 DLOOP
 C83F EEXIT
 C83F EP0INT
 CCF2 EXIT3
 CCF3 EXIT4
 CCA3 G3WAIT
 2B HBASH
 C078 HNDX
 C021 IN1
 C86F INEX1
 C098 INX
 C0EFC JSSCALE
 CABA KLOOP
 CAF3 L00P2
 C095A LORES
 C43B MPAGE
 CA67 NO
 C8DC NOSWITCH
 C864 OFFSET
 C6B8 OFFYL
 C94E OUT1
 CE7F PARAM
 C82D PRCHK
 CC19 RDLOOP
 C083 RESLP
 CD24 ROUTIN
 C53B SCALH
 C057 SHIRES
 C054 SPAG1
 CE44 STRIN
 CCAD SYNTAX
 C085 TEMXL
 C080 TEM
 C022 XFLH
 C023 YFLL
 2B HBASH
 C021 IN1
 C021 XFLL
 C025 TEMXL
 C028 TEMY
 C099 NFLAG
 C0A2 OREGH
 C0A3 COUNT
 C53B SCALH
 C038 OFFXH
 C073 OFFYH
 C050 SGR
 C054 SPAG1
 C080 DEV1
 C800 OTHROM
 C83F EP0INT
 C883 INEX2
 C8C6 ASCEX
 C0902 POINT
 C936 TEXT
 C962 LOR1
 C99E OUTSIDE
 C9D0 OUT

C9D8 CTRCHK
 CA2B HPOSN1
 CASF YES
 CAB6 FLOOP
 CAAF JLOOP
 C0AC1 RTN
 CAF2 POS
 C82C FEXIT
 C858 TWOCOM
 C8C6 OFFS1
 CC0C C4SKP
 CC42 READTAB
 CC79 ATST
 CCA1 GWAIT
 C8C0 SYNT1
 CCF3 EXIT4
 CD0E ALPH1
 CD48 NOCR6
 C055 EMSQ1
 CD89 BROUT
 CDCB DIG3
 CDFA CROUT3
 CE44 STRIN
 CE90 DEFAULT
 CEE2 DEF3
 C0EFC JSSCALE
 FF5B IORTS

C9DE LOCLC
 CA2C HPOSN2
 CA67 NO
 CAB9 ASKIP
 CAB2 CSKIP
 CA12 OFFDIV
 CAF3 L00P2
 C082D DEXIT
 C864 OFFSET
 C8D0 SWCHK
 CC0E ZDLOOP
 CC44 ALOOP
 CC7B BTST
 CCA2 G2WAIT
 CC64 SYNT2
 CCF6 PROC1
 CD24 ROUTIN
 CD4C EXIT5
 CD55 EMSQ1
 CD90 DIGLP
 CDD9 DIG4
 CE0C EXIT1
 CE5D TABL
 CE8E STMODE
 CEEA DEF4
 CFFF ROMSW

C9F0 BASCLC
 CA39 CLOOP
 CA6A ASCON
 CA9D HLOOP
 CABA KLOOP
 CAD1 DIVIDE
 C80C L00P3
 C83F EEXIT
 C870 SCALE
 C8DC NOSWITCH
 CC19 RDLOOP
 CC5E A1LOOP
 CC83 RESLP
 CCA3 G3WAIT
 CCE4 CHOUT
 CCFB PROC2
 CD31 ROUT1
 CD4F ERR3
 CD67 AROUT
 CDA2 DIG2
 CDEA CROUT
 CE3E PRINT
 CE6E ADR
 CED2 DEF1
 C0EF6 JWINCHK
 FDED COUT

CA0C C2SKP
 CA4C WINCHK
 CAB0 POSIT
 CAA0 BSKIP
 CABD DSKIP
 CAD7 L00P1
 C0B1F L00P4
 C837 E1EXIT
 C889 MREAD
 CBF4 M1READ
 CC41 OFFSC
 CC64 DLOOP
 CC8E AMOVE
 CCA0 SYNTAX
 CCF2 EXIT3
 CD0C C5SKP
 CD3D ROUT2
 CD50 ERR1
 CD7C NOROL
 CDBB DIG1
 CDF7 CROUT2
 CE41 EXIT2
 CE7F PARAM
 CEDA DEF2
 C0EF9 JHREAD
 FE93 SETVID

QUICK-DRAW

```

0000      1          PAGE
0000      2 *****
0000      3 *
0000      4 * COPYRIGHT 1979
0000      5 *
0000      6 * APPLE COMPUTER INC.
0000      7 *
0000      8 * CUPERTINO CALIFORNIA
0000      9 *
0000     10 * ALL RIGHTS RESERVED
0000     11 *
0000     12 *****
0000     13 * WRITTEN JAN 1979
0000     14 * BY JOHN A
0000     15 * APPLE COMPUTER
0000     16 * SYSTEMS SOFTWARE
0000     17 *****
0000     18 * WINDOCK CORRECTED APR 25, 1979
0000     19 * BY JOHN A
0000     20          PAGE
0000     21 *****
0000     22 *
0000     23 * BITPAD TO APPLESOFTII *
0000     24 * INTERFACE ROUTINE *
0000     25 * THIS ROUTINE MAKES *
0000     26 * IT POSSIBLE TO CALL *
0000     27 * THE BITPAD AT HIGH *
0000     28 * SPEED AND FETCH THE *
0000     29 * POINTS DRAWN DIRECTLY *
0000     30 * INTO APPLESOFT DATA *
0000     31 * ARRAYS AT MAXIMUM *
0000     32 * SPEED. ARRAYS MUST BE *
0000     33 * DIMENSIONED BEFORE *
0000     34 * CALLING THIS ROUTINE *
0000     35 * THEY ARE X% AND Y% *
0000     36 * IN ADDITION NX AND DX *
0000     37 * MUST ALSO BE ALLOCATED *
0000     38 *****
0000     39          ORG $C00
0000     40          OBJ $2000
0000     41 *****
0000     42 *PAGE ZERO USAGE *
0000     43 VARPNT      EQU $83
0000     44 VARNAM      EQU $81
0000     45 LOWTR      EQU $9B
0000     46 TTTTAB      EQU $67
0000     47 VARTAB      EQU $69
0000     48 ARYTAB      EQU $6B
0000     49 STREND      EQU $6D
0000     50 ARYPNT      EQU $94
0000     51 *FAC EQU $9D TO A3
0000     52 DELTA      EQU $9D
0000     53 INDX      EQU $9E
0000     54 NADRS      EQU $A0
0000     55 MAXN      EQU $A2
0000     56 *ARG EQU $A5 TO A8
0000     57 TICFLQ      EQU $A5
0000     58 XVPTR      EQU $A6
0000     59 YVPTR      EQU $A8
0000     60 TMAXY      EQU $AA          ;TEMP MAX Y COORD LIMIT
0000     61 * TMAXY =MIN(MIXED MODE*160,MAXY)
0000     62 XQL      EQU $E0          ;PRIOR X-COORD SAVE
0000     63 XQH      EQU $E1          ;PRIOR X-COORD SAVE HI
0000     64 YQ      EQU $E2          ;PRIOR Y-COORD SAVE
0000     65 ERRFLQ      EQU $DB
0000     66 REMSTR      EQU $FB
0000     67 ERRNUM      EQU $DE
0000     68 *****
0000     69 *ENTRY POINTS USED *

```

```

0C00      70 HILIN      EQU $F53A          ; REAL HILIN ENTRY
0C00      71 HPL0T      EQU $F457          ; REAL HPL0T ENTRY
0C00      72 CHRGET      EQU $00B1          ; NEXT PGM CHAR
0C00      73 CHRGET      EQU $00B7
0C00      74 CRDO      EQU $DAFB
0C00      75 OUTGST      EQU $DB5A
0C00      76 OUTDD      EQU $DB5C
0C00      77 ISLETC      EQU $E07D
0C00      78 TYPERR      EQU $D42A
0C00      79 HNDLERX      EQU $F2EF
0C00      80 WAIT      EQU $FCAB          ; MON A WAIT
0C00      81          PAGE
0C00      82 *****
0C00      83 * DEVICE ADDRESSES
0C00      84 SPKR      EQU $C030          ; TOGGLES APPLE SPKR
0C00      85 STXT      EQU $C051          ; SET TEXT MODE!
0C00      86 SPAG1      EQU $C054
0C00      87 *****
0C00      88 * BITPAD INTERFACE EQU *
0C00      89 *****
0C00      90 TEM      EQU $2B0          ; RETURN FLAG LOC
0C00      91 * HI NIBBLE =1 FOR SCALED RESULTS
0C00      92 * LO NIBBLE 0=PEN DOWN
0C00      93 * 1=PEN LIFT
0C00      94 * 2 PEN FALL
0C00      95 * 3=PEN UP
0C00      96 XFLL      EQU $2B1          ; X-COORD LO UNSCALED
0C00      97 XFLH      EQU $2B2          ; X-COORD HI UNSCALED
0C00      98 YFLL      EQU $2B3          ; Y-COORD LO UNSCALED
0C00      99 YFLH      EQU $2B4          ; Y-COORD HI UNSCALED
0C00     100 TEMXL      EQU $2B5          ; X-COORD LO SCALED
0C00     101 TEMX      EQU $2B6          ; X-COORD HI SCALED
0C00     102 TEMYL      EQU $2B7          ; Y-COORD LO SCALED
0C00     103 TEMY      EQU $2B8          ; Y-COORD HI SCALED
0C00     104 SSM1      EQU $29A          ; LO INDIRECT ADDR
0C00     105 SAVSLOT      EQU $29B
0C00     106 RTNCD      EQU $2BC          ; =700 BITSOFT RTN CODE
0C00     107 PAGE      EQU $308          ; +CN SCREEN MODE
0C00     108 * HI BIT=1 MEANS SCALE DATA
0C00     109 * 40 = HIRES PG2
0C00     110 * 20 = HIRES PG1
0C00     111 HIRES1      EQU $20
0C00     112 * 08 = TEXT PG2
0C00     113 * 04 = TEXT PG1
0C00     114 * 02 = LORES PG2
0C00     115 * 01 = LORES PG1
0C00     116 * 42 = HIRES PG2 MIXED
0C00     117 * 21 = HIRES PG1 MIXED
0C00     118 * 0A = LORES PG2 MIXED
0C00     119 * 05 = LORES PG1 MIXED
0C00     120 MXYVALU      EQU $60          ; FOR NORMAL APPLE
0C00     121 MSLOT      EQU $7FB
0C00     122 *****
0C00     123 * BITPAD ENTRY POINTS *
0C00     124 *****
0C00     125 POINT      EQU $C102
0C00     126 MREAD      EQU $CEFA
0C00     127 WINCHK      EQU $CEFA
0C00     128 SCALE      EQU $CEFC
0C00     129          PAGE
0C00     130 *****
0C00     131          JMP BITSOFT
0C00     132          JMP FINDVAR          ;FIND VARIABLE UTILITY ENTR
0C00     133          JMP FINDARY          ;FIND ARRAY UTILITY ENTRY
0C00     134 *****
0C00     135 VNAMTAB      EQU *
0C00     136 DLTANAM      DFB $C4,$B0          ; DX
0C00     137 NDXNAM      DFB $CE,$B0          ; NX
0C00     138 XVNAM      DFB $DB,$B0          ; XX
0C00     139 YVNAM      DFB $D9,$B0          ; YY
0C00     140 DNAME      EQU DLTANAM-VNAMTAB
0C00     141 NNAME      EQU NDXNAM-VNAMTAB
0C00     142 XNAME      EQU XVNAM-VNAMTAB

```



```

0C11: 143 YNAME      EGU YVNAME-VNAME TAB
0C11: 00          144 MINXL     DFB $00
0C12: 00          145 MINX      DFB $00
0C13: 18          146 MAXXL     DFB 24
0C14: 01          147 MAXX      DFB 1
0C15: 00          148 MINY      DFB $0
0C16: C0          149 MAXY      DFB 192
0C17: 03 4F D0    150 CTABLE     DFB $03,$4F,$D0
0C1A: 19 52 C9    151          DFB $19,$52,$C9
0C1D: 07 48 D4    152          DFB $07,$48,$D4
0C20: 20 41 D0    153          DFB $20,$41,$D0
0C23: 10 4C C5    154          DFB $10,$4C,$C5
0C26: 20 43 CF    155          DFB $20,$43,$CF
0C29: 00 50 D5    156          DFB $00,$50,$D5
0C2C: 14 45 D2    157          DFB $14,$45,$D2
0C2F: 37 79       158          DFB $37,$79
0C31: 159         159          PAGE
0C31: 160         160 *****
0C31: 161 * BITPAD MESSAGES
0C31: 20 44 41
0C34: 50 54 49
0C37: C2
0C38: 44 45 4C    162 BITMSG      DCI *
0C38: 54 41 20          DAPTIB"
0C3E: 53 49 5A
0C41: C5
0C42: 44 45 4C    163 DLTASIZ      DCI "DELTA
0C45: 54 41 20          SIZE"
0C48: 55 4E 44
0C48: 45 46 49
0C4E: 4E 45 C4    164 DLTADef      DCI "DELTA
0C51: 49 4E 44          UNDEFINED"
0C54: 45 58 20
0C57: 55 4E 44
0C5A: 45 46 49
0C5D: 4E 45 C4    165 INDXDEF      DCI "INDEX
0C60: 41 52 52          UNDEFINED"
0C63: 41 59 20
0C66: 44 49 4D
0C69: 43 4E 53
0C6C: 49 4F CE    166 ARYDEF      DCI "ARRAY
0C6F: 42 41 44          DIMENSION"
0C72: 20 53 55
0C75: 42 53 43
0C78: 52 49 50
0C7B: D4
0C7C: 4E 4F 54    167 NGTMAXN      DCI "BAG
0C7F: 20 49 4E          SUBSCRIPT"
0C82: 20 53 43
0C85: 41 4C 45
0C88: 44 20 48
0C8B: 49 52 45
0C8E: 53 20 4D
0C91: 4F 44 C5    168 NOTHGR      DCI "NOT
0C94: 169 XDLTASIZ      EGU DLTASIZ-BITMSG+40 :THIS BIASES RTN CDS BY 4
0C94: 170 XDLTADef      EGU DLTADef-BITMSG+40 :THIS BIASES RTN CDS BY 4
0C94: 171 XINDXDEF      EGU INDXDEF-BITMSG+40 :THIS BIASES RTN CDS BY 4
0C94: 172 XARYDEF      EGU ARYDEF-BITMSG+40 :THIS BIASES RTN CDS BY 4
0C94: 173 XNRANGE      EGU NGTMAXN-BITMSG+40 :THIS BIASES RTN CDS BY 4
0C94: 174 XNOTHGR      EGU NOTHGR-BITMSG+40 :THIS BIASES RTN CDS BY 4
0C94: 175          PAGE
0C94: 176 *****
0C94: 177 * BEGIN CODE *
0C94: 178 *****
0C94: 179 BITSOFT      EGU *
0C94: 180          JSR USRNAME
0C97: A2 00          181 NONAMES      LDX #NAME
0C99: 20 E4 0E          182          SET UP VARNA
0C9C: 20 58 0F          183          GO FIND DELTA
0C9F: 00 33          184          BCS DLTACK
0CA1: A2 39          185          LDX #DLTADef
0CA3: 24 D0          186 BITPERR      EGU *
0CA3: 187          BIT ERRFLG

```

```

0CA5: 10 08          188          BPL DOERR
0CA7: 86 DE          189          STX ERRNUM
0CA9: A6 FB          190          LDX REMSTK
0CAB: 9A            191          TXS
0CAC: 4C EF F2      192          JMP HNDLERX
0CAF: 193          193 DOERR      EGU *
0CAF: A0 54 C0      194          LDA SPAG1
0CB2: A0 51 C0      195          LDA STEXT
0CB5: 20 FB DA      196          JSR CRDO
0CB8: 20 5A D8      197          JSR OUTGST
0CB8: A0 06          198          LDY #6
0CB8: 89 31 0C      199 BITPD      LDA BITMSG.Y
0CC0: 20 5C D8      200          JSR OUTDO
0CC3: 88            201          DEY
0CC4: 10 F7          202          BPL BITPD
0CC6: 8D 09 0C      203 BMSGPL      LDA BITMSG-40,X
0CC9: 48            204          PHA
0CCA: 20 5C D8      205          JSR OUTDO
0CCD: E8            206          INX
0CCE: 68            207          PLA
0CCF: 10 F5          208          BPL BMSGPL
0CD1: 4C 2A D4      209          JMP TYPERR
0CD4: 210          210 DLTAOK      EGU *
0CD4: A0 00          211          LDY #0
0CD6: 81 83          212          LDA (VARPNT),Y
0CD8: 84 A5          213          STY TICFLG
0CDA: 18            214          CLC
0CDB: 10 08          215          BPL CHKDVALU
0CDD: 38            216          SEC
0CDE: 85 A5          217          STA TICFLG
0CE0: 49 FF          218          EOR #$FF
0CE2: F0 04          219          BEQ CHKDVALU
0CE4: A2 2F          220 DSIZERR      LDX #XDLTASIZ
0CE6: D0 D0          221          BNE BITPERR
0CEB: 222          222 CHKDVALU      EGU *
0CEB: C8            223          INY
0CE9: 81 83          224          LDA (VARPNT),Y
0CE9: 90 04          225          BCC NOCOMPL
0CED: 49 FF          226          EOR #$FF
0CEF: 69 00          227          ADC #0
0CF1: 30 F1          228 NOCOMPL      BMI DSIZERR
0CF3: F0 EF          229          BEQ DSIZERR
0CF5: 85 9D          230          STA DELTA
0CF7: A2 02          231          LDX #NAME
0CF9: 20 E4 0E          232          JSR SETNAME
0CFC: 20 58 0F          233          JSR FINDVAR
0CFF: D0 04          234          BCS NTHERR
0D01: 235          235          PAGE
0D01: A2 48          236 NDEFERR      LDX #XINDXDEF
0D03: D0 9E          237          BNE BITPERR
0D05: 238          238 NTHERR      EGU *
0D05: A0 00          239          LDY #0
0D07: 81 83          240          LDA (VARPNT),Y
0D09: 30 35          241          BMI NRNGERR
0D0B: 85 9F          242 IXOK          STA INDX+1
0D0D: C8            243          INY
0D0E: 81 83          244          LDA (VARPNT),Y
0D10: 85 9E          245          STA INDX
0D12: A5 83          246          LDA VARPNT
0D14: 85 A0          247          STA NADRS
0D16: A5 84          248          LDA VARPNT+1
0D18: 85 A1          249          STA NADRS+1
0D1A: A2 04          250          LDX #NAME
0D1C: 20 E4 0E          251          JSR SETNAME
0D1F: 20 27 0F          252          JSR FINDARY
0D22: D0 05          253          BCS XTHERR
0D24: A2 57          254 ARYERR      JMP BITPERR
0D26: 4C A3 0C          255 ERRJMP      EGU *
0D29: 256          256 XTHERR      LDY #4
0D29: A0 04          257          LDA (VARPNT),Y
0D2B: 81 83          258          CMP #1
0D2D: C9 01          259          BNE ARYERR
0D2F: D0 F3          260

```



```

0031 C0 261 XDIMBOK INY / Y=5
0032 C9 262 INY / Y=6
0033 00 263 SEC
0034 01 00 264 LDA (VARPNT),Y
0036 05 00 265 SBC INDX
0038 AA 266 TAX / SAVE LOW RESULT
0039 00 267 DEY / Y=5
003A 01 00 268 LDA (VARPNT),Y
003C 03 00 269 SBC INDX+1
003E 00 04 270 BCS NGTXROWS
0040 A2 04 271 LDX #XNRANGE
0042 00 02 272 BNE ERRJMP
0044 00 FA 273 DMI NRRNGERR
0046 00 04 274 BNE NISOK
0048 E0 02 275 CPX #2
004A 00 F4 276 BCC NRRNGERR
004C 01 00 277 LDA (VARPNT),Y / GET MAX
004E 05 A3 278 STA MAXN+1
0050 C8 279 INY / Y=6
0051 01 00 280 LDA (VARPNT),Y / GET LO
0053 05 A2 281 STA MAXN
0055 A0 00 282 LDA VARPNT
0057 18 283 CLC
0059 09 07 284 ADC #7 / CALC BASE ADDRESS
005A 05 A6 285 STA XVPTR / SET X BASE
005C A6 04 286 LDX VARPNT+1 / GET HI BASE
005E 00 01 287 BCC #+3 / NO CARRY FROM ADD
0060 E8 288 INX
0061 06 A7 289 STX XVPTR+1 / X BASE COMPLETE
0063 290 PAGE + =6
0065 20 E4 0E 292 JSR SETNAME
0068 20 27 0F 293 JSR FINDARY
006B 00 07 294 BCC ARYERR
006D A0 04 295 LDY #4 / Y THERE
006F 01 00 296 LDA (VARPNT),Y
0071 C9 01 297 CMP #1
0073 00 AF 298 BNE ARYERR
0075 C8 299 INY
0076 C8 300 INY / POINT TO NROWSL
0077 01 00 301 LDA (VARPNT),Y
0079 AA 302 TAX
007A C5 A2 303 CMP MAXN
007C 88 304 DEY
007D 01 00 305 LDA (VARPNT),Y
007F E5 A3 306 SBC MAXN+1
0081 00 1C 307 BCS YSIZOK / IS YSIZE>XSIZE?
0083 06 A2 308 STX MAXN / YSIZE>XSIZE
0085 01 00 309 LDA (VARPNT),Y / GET HI BACK
0087 05 A3 310 STA MAXN+1
0089 38 311 SEC
008A A5 A2 312 LDA MAXN
008C E5 0E 313 SBC INDX
008E AA 314 TAX
008F A5 A3 315 LDA MAXN+1
0091 E5 0F 316 SBC INDX+1
0093 00 02 317 BCS YROWSGTN
0095 00 A9 318 BCC NRRNGERR
0097 00 A7 319 DMI NRRNGERR
0099 00 04 320 BNE YSIZOK
009B E0 02 321 CPX #2 / AT LEAST TWO MORE ARRAY E
009D 00 A1 322 BCC NRRNGERR / YSIZOK
009F 323 EQU *
00A1 18 324 LDA VARPNT
00A2 09 07 325 CLC
00A4 05 A8 326 ADC #7
00A6 A6 04 327 STA YVPTR
00A8 00 01 328 LDX VARPNT+1
00AA E8 329 BCC #+3
00AB 06 A9 330 INX
00AD 06 0E 331 STX YVPTR+1
00AF 26 0F 332 ASL INDX
333 ROL INDX+1 INDX=INDX*2

```

```

00B1 A5 A8 334 LDA YVPTR
00B3 A6 A9 335 LDX YVPTR+1
00B5 20 F0 0E 336 JSR ADDINX
00B8 04 A8 337 STY YVPTR
00BA 05 A9 338 STA YVPTR+1
00BC A5 A6 339 LDA XVPTR
00BE A6 A7 340 LDX XVPTR+1
00C0 20 F0 0E 341 JSR ADDINX
00C3 04 A6 342 STY XVPTR
00C5 05 A7 343 STA XVPTR+1
00C7 46 0F 344 LSR INDX+1
00C9 06 0E 345 ROR INDX / INDX=INDX/2
00CB 346 PAGE
00CD A9 00 347 LDA #0
00CF 00 0C 02 348 STA RTNCD
00D1 AC 00 02 349 LDY SAVSLOT
00D3 08 350 TYA / PUT IN A
00D5 38 351 SEC / FOR SUBTRACT
00D7 E9 C1 352 SBC #C1 / IT VALID SLOT?
00D9 C9 07 353 CMP #7
00DB 00 08 354 BCS BADMODE / SLOT NOT INITED!!!
00DD 09 08 03 355 LDA PAGE,Y
00DE 10 06 356 BPL BADMODE
00E0 29 0F 357 AND #07F
00E2 C9 20 358 CMP #HIRES1
00E4 00 05 359 BCS MODEOK
00E6 A2 73 360 BADMODE LDY #XNOTHGR
00E8 AC A3 0C 361 JMP DITPERR
00EA 29 03 362 MODEOK AND #3
00EC AC 16 0C 363 LDY MAXY
00EE AA 364 TAX / IS IT MIXED MODE?
00F0 F0 06 365 BEQ NOTMIXD
00F2 C0 A0 366 CPY #MAXVALU
00F4 00 02 367 BCC NOTMIXD
00F6 A0 A0 368 LDY #MAXVALU
00F8 04 AA 369 NOTMIXD STY MAXY
00FA A2 02 370 LDY #POINT
00FC 0E 0A 02 371 STX SAVSLOT-1
00FE 20 E1 0E 372 JSR JSRINDRCT
0100 AD 00 02 373 LDA TEM
0102 29 03 374 AND #3
0104 C9 03 375 CMP #3 / IS PEN UP?
0106 F0 0A 376 BEQ KYBDXIT / YES, HE HIT KEYBD
0108 20 F8 0E 377 JSR WINDOCHM / IS IT ON SCREEN?
010A 00 06 378 BCS ONSCRN / YEP
010C A9 01 379 YTOOBIG / PEN OFF SCREEN EXIT
010E 00 0C 02 380 STRTNCN
0110 00 0C 02 381 KYBDXIT
0112 60 382 ONSCRN
0114 AD 07 02 383 EQU *
0116 AE 05 02 384 LDA TEMYL
0118 AC 06 02 385 LDX TEMXL
011A 20 57 F4 386 LDY TEMX
011C 24 A5 387 JSR HPLDT
011E 20 0B 388 EQU *
0120 24 A5 389 BIT TICFLG
0122 30 0B 390 BMT NOTICK
0124 AD 30 C0 391 EQU *
0126 A9 0F 392 LDA SPKR
0128 20 AB FC 393 LDA #0F
012A AD 30 C0 394 JSR WAIT
012C AD 30 C0 395 LDA SPKR / TICK IT
012E AD 06 02 396 EQU *
0130 A0 00 397 LDA TEMX
0132 01 A6 398 LDY #0
0134 AD 0B 02 399 STA (XVPTR),Y
0136 01 A8 400 LDA TEMY
0138 C8 401 STA (YVPTR),Y / STORE Y VALU
013A AD 05 02 402 INY
013C 01 A6 403 LDA TEMXL / STORE X VALUE
013E AD 07 02 404 STA (XVPTR),Y
0140 01 A8 405 STA (YVPTR),Y

```



```

0E49: 406 PAGE
0E49: A2 01 407 LDX #1
0E4B: E6 A6 408 INC XVPTR
0E4D: D0 02 409 BNE **+4
0E4F: E6 A7 410 INC XVPTR+1
0E51: CA 411 DEX
0E52: F0 F7 412 BEQ XVINC
0E54: A2 01 413 LDX #1
0E56: E6 A8 414 YVINC INC YVPTR
0E58: D0 02 415 BNE **+4
0E5A: E6 A9 416 INC YVPTR+1
0E5C: CA 417 DEX
0E5D: F0 F7 418 BEQ YVINC
0E5F: E6 9E 419 INC INDX
0E61: D0 02 420 BNE **+4
0E63: E6 9F 421 INC INDX+1
0E65: 422 * (Y=1 FROM ABOVE)
0E65: A5 9E 423 LDA INDX
0E67: 91 A0 424 STA (NADRS), Y
0E69: 88 425 DEY
0E6A: A5 9F 426 LDA INDX+1
0E6C: 91 A0 427 STA (NADRS), Y
0E6E: A5 9E 428 LDA INDX
0E70: C5 A2 429 CMP MAXN
0E72: A5 9F 430 LDA INDX+1
0E74: E5 A3 431 SBC MAXN+1
0E76: 90 05 432 BCC MORPTS
0E78: A9 03 433 LDA #3
0E7A: 4C 13 0E 434 JMP STRTNC
0E7D: AE 9B 02 435 MORPTS LDX SAVSLOT
0E80: BE F8 07 436 STX MSLT
0E83: 437 WAITLP EQU *
0E83: 20 F9 CE 438 JSR MREAD
0E86: AD 80 02 439 LDA TEM
0E89: 29 03 440 AND #3
0E8B: F0 05 441 BEQ PENDOWN
0E8D: A9 02 442 LDA #2
0E8F: 4C 13 0E 443 JMP STRTNC
0E92: A2 03 444 LDX #3
0E94: B0 B1 02 445 MVLV LDA XPLL, X
0E97: 9D B5 02 446 STA TEMXL, X
0E9A: CA 447 DEX
0E9D: 10 F7 448 BPL MVLV
0E9D: AE 9B 02 449 LDX SAVSLOT
0EA0: 20 FC CE 450 JSR SCALE
0EA3: 20 F8 0E 451 JSR WINDOCHK
0EA6: B0 03 452 BCS CKDLTA
0EA8: 4C 11 0E 453 JMP YTOOSIG
0EA8: 454 CKDLTA EQU *
0EA8: AD B7 02 455 LDA TEMYL
0EAB: 38 456 SEC
0EAF: E5 E2 457 SBC Y0
0EB1: B0 04 458 BCS CKYDLTA
0EB3: 49 FF 459 EOR **FF
0EB5: 69 01 460 ADC #1
0EB7: C5 9D 461 CKYDLTA CMP DELTA
0EB9: B0 17 462 BCS PLOTSEQ
0EBB: 38 463 SEC
0EBC: AD B5 02 464 LDA TEMXL
0EBF: E5 E0 465 SBC XOL
0EC1: AA 466 TAX
0EC2: AD B6 02 467 LDA TEMX
0EC5: E5 E1 468 SBC XOH
0EC7: BA 469 TXA
0ECB: B0 04 470 BCS CKXDLTA
0ECA: 49 FF 471 EOR **FF
0ECC: 69 01 472 ADC #1
0ECE: 473 CKXDLTA EQU *
0ECE: C5 9D 474 CMP DELTA
0ED0: 90 B1 475 BCC WAITLP
0ED2: AD B5 02 476 PLOTSEQ LDA TEMXL
0ED5: AE B6 02 477 LDX TEMX
0ED8: AC B7 02 478 LDY TEMYL

```

```

0ED8: 20 3A F5 479 JSR HILIN
0EDE: 4C 23 0E 480 JMP MAINLP
0EE1: 481 *****
0EE1: 482 * MY SUBROUTINES
0EE1: 483 *****
0EE1: 6C 9A 02 484 JSRINDRCT JMP (SAVSLOT-1)
0EE4: 485 *****
0EE4: BD 09 0C 486 SETNAME LDA VNAMTAB, X
0EE7: B5 B1 487 STA VARNAM
0EE9: E8 488 INX
0EEA: BD 09 0C 489 LDA VNAMTAB, X
0EED: B5 B2 490 STA VARNAM+1
0EEF: 60 491 RTS
0EFO: 492 *****
0EFO: 18 493 ADDINX CLC
0EF1: 65 9E 494 ADC INDX
0EF3: AB 495 TAY
0EF4: BA 496 TXA
0EF5: 65 9F 497 ADC INDX+1
0EF7: 60 498 RTS
0EF8: 499 *****
0EF8: AD B5 02 500 WINDOCHK LDA TEMXL
0EF8: AA 501 TAX
0EFC: CD 11 0C 502 CMP MINXL
0EFF: AD B6 02 503 LDA TEMX
0F02: AB 504 TAY
0F03: ED 12 0C 505 SBC MINX
0F06: 90 0B 506 BCC WCHARTS
0F08: BA 507 TXA
0F09: CD 13 0C 508 CMP MAXXL
0F0C: 98 509 TYA
0F0D: ED 14 0C 510 SBC MAXX
0F10: 90 02 511 BCC XINSIDE
0F12: 18 512 CLC
0F13: 60 513 RTS
0F14: AD B8 02 514 XINSIDE LDA TEMY
0F17: D0 F9 515 BNE OUTSIDE
0F19: AD B7 02 516 LDA TEMYL
0F1C: CD 15 0C 517 CMP MINY
0F1F: 90 F2 518 BCC WCHARTS
0F21: C5 AA 519 CMP TMAXY
0F23: B0 ED 520 BCS OUTSIDE
0F25: 38 521 SEC
0F26: 60 522 RTS
0F27: 523 PAGE
0F27: 524 *****
0F27: A6 B8 525 FINDARY LDX ARYTAB
0F29: A3 6C 526 LDA ARYTAB+1
0F2B: B6 98 527 FNDLPA STX LOWTR
0F2D: B5 9C 528 STA LOWTR+1
0F2F: C5 6E 529 CMP STREND+1
0F31: D0 04 530 BNE FNDFDV
0F33: E4 6D 531 CPX STREND
0F35: F0 3C 532 BEQ NOTFND
0F37: A0 00 533 FNDFDV LDY #0
0F39: B1 98 534 LDA (LOWTR), Y
0F3B: C8 535 INY
0F3C: C5 B1 536 CMP VARNAM
0F3E: D0 06 537 BNE NXTARY
0F40: A5 B2 538 LDA VARNAM+1
0F42: D1 98 539 CMP (LOWTR), Y
0F44: F0 0E 540 BEQ GOTARY
0F46: C8 541 NXTARY INY
0F47: B1 98 542 LDA (LOWTR), Y
0F49: 18 543 CLC
0F4A: 65 98 544 ADC LOWTR
0F4C: AA 545 TAX
0F4D: C8 546 INY
0F4E: B1 98 547 LDA (LOWTR), Y
0F50: 65 9C 548 ADC LOWTR+1
0F52: 90 D7 549 BCC FNDLPA
0F54: A9 00 550 GOTARY LDA #0
0F56: F0 2D 551 BEQ ADJVPTR

```



```

OF5B: 552 PAGE
OF5B: 553 *****
OF5B: A3 69 554 FINDVAR LDA VARTAB
OF5A: A6 6A 555 LDX VARTAB+1
OF5C: A0 00 556 LDY #0
OF5E: 86 9C 557 FNDLPX STX LOWTR+1
OF60: 85 9B 558 FINDLP STA LOWTR
OF62: E4 6C 559 CPX ARYTAB+1
OF64: D0 04 560 BNE LOPFN
OF66: C5 6B 561 CMP ARYTAB
OF68: F0 29 562 BEQ NOTFND
OF6A: A3 81 563 LOPFN LDA VARNAM
OF6C: D1 9B 564 CMP (LOWTR),Y
OF6E: D0 0B 565 BNE NOTIT
OF70: A5 82 566 LDA VARNAM+1
OF72: C8 567 INY
OF73: D1 9B 568 CMP (LOWTR),Y
OF75: F0 0C 569 BEQ FOUNDS
OF77: 88 570 DEY
OF79: 18 571 NOTIT CLC
OF7A: A5 9B 572 LDA LOWTR
OF7B: 69 07 573 ADC #7
OF7D: 90 E1 574 BCC FINDLP
OF7F: E8 575 INX
OF80: D0 0C 576 BNE FNDLPX
OF82: 00 577 BRK
OF83: A9 02 578 FOUNDS LDA #2
OF85: 18 579 ADJVPR CLC
OF86: 65 9B 580 ADC LOWTR
OF88: A4 9C 581 LDY LOWTR+1
OF8A: 90 01 582 BCC ADJDONE
OF8C: C8 583 INY
OF8D: 85 83 584 ADJDONE STA VARPNT
OF8F: 84 84 585 STY VARPNT+1
OF91: 38 586 SEC
OF92: 60 587 RTS
OF93: 18 588 NOTFND CLC
OF94: 60 589 RTS
OF95: 590 *****
OF95: 591 * GET USR NAMES FROM *
OF95: 592 * HIS CALL LINE AND *
OF95: 593 * USE THEM INSTEAD *
OF95: 594 * OF THE DEFAULTS *
OF95: 595 * THE USER ENTERS *
OF95: 596 * THE NAMES IN FIXED *
OF95: 597 * POSITIONAL ORDER *
OF95: 598 * AS FOLLOWS *
OF95: 599 * D%,N%,X%,Y% *
OF95: 600 *****
OF95: 601 PAGE
OF95: 602 USRNAMS EQU *
OF95: A2 07 603 LDX #7
OF97: B0 F8 0F 604 DFLT5 LDA DEFALT,X
OF9A: 90 09 0C 605 STA VNAMTAB,X
OF9D: CA 606 DEX
OF9E: 10 F7 607 BPL DFLT5
OFA0: 20 B7 00 608 JSR CHRGET
OFA3: D0 01 609 BNE **3
OFA5: 60 610 RTS
OFA6: 20 7D E0 611 JSR ISLETC
OFA9: 90 18 612 BCC GETLTR
OFAB: 20 E7 0F 613 SET1ST JSR STORIT
OFAE: F0 31 614 BEQ UNAMRTS
OFB0: 20 F0 0F 615 ISTAIL JSR MYCHGET
OFB3: 90 1E 616 BCC SET2ND
OFB5: 20 7D E0 617 JSR ISLETC
OFB8: B0 19 618 BCS SET2ND
OFBA: C9 2C 619 CMP #*2C
OFBC: D0 F2 620 BNE ISTAIL
OFBE: E8 621 NXTX INX
OFBF: E0 07 622 CPX #7
OFC1: B0 1E 623 BCS UNAMRTS
OFC3: 20 F0 0F 624 GETLTR JSR MYCHGET

```

```

; X,A SET UP
; DO CARRY TO HI
; X,A SET NOW
; INSURANCE!

```

```

; GET NEXT CHR
; EXIT, END OF STMT
; A LETTER?
; NO, IGNORE IT
; YES - USE

; GET NEXT
; DIGIT OK
; A LETTER?
; YES, USE
; NO, A COMMA?
; NOT COMMA, IGNORE
; A COMMA SAYS NO 2ND
; AM I DONE?
; YEP
; FIND A LTR

```

```

OFC6: 20 7D E0 625 JSR ISLETC
OFC9: B0 E0 626 BCS SET1ST
OFCB: C9 2C 627 CMP #*2C
OFCD: D0 F4 628 BNE GETLTR
OFCF: E8 629 INX
OFD0: B8 630 CLV
OFD1: 30 E8 631 BVC NXTX
OFD3: 20 E7 0F 632 SET2ND JSR STORIT
OFD6: F0 09 633 BEQ UNAMRTS
OFD8: 20 F0 0F 634 SCANC JSR MYCHGET
OFDB: C9 2C 635 CMP #*2C
OFDD: D0 F9 636 BNE SCANC
OFDF: F0 E2 637 BEQ GETLTR
OFE1: 20 F0 0F 638 UNAMRTS JSR MYCHGET
OFE4: D0 F8 639 BNE UNAMRTS
OFE6: 60 640 RTS
OFE7: E8 641 STORIT INX
OFE8: 09 80 642 ORA #*80
OFEA: 9D 09 0C 643 STA VNAMTAB,X
OFED: E0 07 644 CPX #7
OFEF: 60 645 RTS
OFF0: 20 B1 00 646 MYCHGET JSR CHRGET
OFF3: D0 02 647 BNE **4
OFF5: 68 648 PLA
OFF6: 68 649 PLA
OFF7: 60 650 RTS
OFFB: C4 80 651 DEFALT DFB #C4,$80
OFFA: CE 90 652 DFB #CE,$80
OFFC: D8 80 653 DFB #D8,$80
OFFE: D9 80 654 DFB #D9,$80

```

*** SUCCESSFUL ASSEMBLY: NO ERRORS

UTILITIES

```

0000: 1 *
0000: 2 *VARIOUS BIT PAD HI-RES ROUTINES
0000: 3 *BY DAVE M. LINCER
0000: 4 *COPYRIGHT APPLE COMPUTER CO.
0000: 5 * JUNE 1979
0000: 6 *
0000: 7 DRQ #6000
0000: 8 DBJ #2000
0000: 9 HBASL EGU #00
0000: 10 HBASH EGU #01
0000: 11 HMASK EGU #02
0000: 12 ZTEM EGU #03
0000: 13 XOL EGU #2FF
0000: 14 XOH EGU #2FE
0000: 15 YO EGU #2FD
0000: 16 FLGL EGU #2FC
0000: 17 FLGH EGU #2FB
0000: 18 XL EGU FLGL
0000: 19 XH EGU FLGH
0000: 20 *
0000: 21 *WHITE PICK OFF
0000: 22 *
0000: 23 WHITE TXA SAVE X,Y AND ZPG
0001: 48 PHA
0002: 98 TYA
0003: 48 PHA
0004: A2 03 LDX #03
0006: B5 00 LDA #00,X
0008: 48 PHA
0009: CA DEX
000A: 10 FA BPL SLP,ZPG
000C: AD FF 02 LDA XOL
000F: BD 55 60 STA PKE1
0012: BD 91 60 STA PKE2
0015: A9 00 LDA #00
0017: BD FD 02 STA YO
001A: BD FF 02 STA XOL
001D: BD FE 02 STA XOH
0020: BD FC 02 STA FLGL
0023: BD FB 02 STA FLGH
0026: 20 C3 60 JSR HPOSN
0029: AE FF 02 LDX XOL
002C: AC FE 02 LDY XOH
002F: 20 ED 60 JSR XPOS
0032: B1 00 LDA (HBASL),Y
0034: 25 02 AND HMASK
0036: B5 03 STA ZTEM
0038: FD 02 BEQ ITZOFF
003A: A9 01 LDA #01
003C: 1B CLC
003D: 6D FC 02 ADC FLGL
0040: BD FC 02 STA FLGL
0043: 29 00 AND #00
0045: 6D FB 02 ADC FLGH
0048: BD FB 02 STA FLGH
004B: AD FC 02 LDA FLGL
004E: C9 02 CMP #02
0050: AD FB 02 LDA FLGH
0053: E9 00 SBC #00
0055: 90 0A BCC LN130
0057: A5 03 LDA ZTEM
0059: D0 06 BNE LN130
005B: 20 02 61 JSR CLER
005E: 4C 73 60 JMP NXTX
0061: AD FC 02 LDA FLGL
0064: 0D FB 02 ORA FLGH
0067: F0 04 BEQ ZFG
0069: A5 03 LDA ZTEM
006B: D0 06 BNE NXTX

```

SAVE X,Y AND ZPG

SAVED
SELF MODIFY
BCC INTO BCS

INIT MY REGS
TO UPPER LEFT
W/NO BITS ON

FIRST TIME
NEXT TIME ONLY
CHANGE Y &
HMASK
DO AN
HSCRN # X,Y
SAVE BIT
??
NOPE
YUP
INC FLG CNTR
BY BIT ON
CARRY FROM LOW ADD
IS ADDED TO FLGH

IS THERE MORE
THAN 2 BITS ON
2 BYTE TEST

CS FOR DEL WHITE
BIT?
ON?
NO, OFF
NEXT HORIZ POS
IS ANY ON?

NO
YES AND IF
Z=1 THEN OK

```

606D: BD FC 02 70 ZFG STA FLGL
6070: BD FB 02 71 STA FLGH
6073: EE FF 02 72 NXTX INC XOL
6076: D0 03 73 BNE TESTX THEN TEST
607B: EE FE 02 74 INC XOH X HIGH
607B: AD FF 02 75 TESTX LDA XOL
607E: C9 1B 76 CMP #01B 2 BYTE TEST
6080: AD FE 02 77 LDA XOH FOR XPOS=
6083: E9 01 78 SBC #001 279 OR
6085: 90 A2 79 BCC BLOOP #11B
6087: AD FC 02 80 LDA FLGL IF OK ELSE
608A: C9 02 81 CMP #002 END OF X LOOP
608C: AD FB 02 82 LDA FLGH MAKE SURE TO
608F: E9 00 83 SBC #000 CATCH LAST
6091: 90 03 84 PKE2 BCC NXTY WHITE OR COLOR
6093: 20 02 61 85 JSR CLER CS FOR DEL CLR
6096: A9 00 86 NXTY LDA #000 CLEAR OUT
6098: BD FC 02 87 STA FLGL RESET X AND
609B: BD FB 02 88 STA FLGH FLAG
609E: BD FF 02 89 STA XOL
60A1: BD FE 02 90 STA XOH
60A4: EE FD 02 91 INC YO
60A7: 20 C3 60 92 JSR HPOSN THEN INC YO
60AA: AD FD 02 93 LDA YO REPOSN
60AD: C9 C0 94 CMP #0C0 AND TEST
60AF: 90 03 95 BCS RET1 FOR Y=191
60B1: 4C 29 60 96 JMP BLOOP WE'RE DONE
60B4: A2 00 97 RET1 NO CONTINUE (LONG BRANCH)
60B6: 6B 98 RLP BRING BACK ALL
60B7: 95 00 99 STA #00,X ZPAGE
60B9: EB 100 INX AND X,Y
60BA: E0 04 101 CPX #004
60BC: D0 FB 102 BNE RLP
60BE: 6B 103 PLA
60BF: AB 104 TAY
60C0: 6B 105 PLA
60C1: AA 106 TAX
60C2: 60 107 RTS
60C3: 108 *
60C3: 109 *HPOSN HRES BIT POSN ROUT
60C3: 110 *READ ABOUT THIS CODE IN UTILITY ROM MANUAL
60C3: 111 *
60C3: AD FD 02 112 HPOSN LDA YO
60C6: AE FF 02 113 LDX XOL
60C9: AC FE 02 114 LDY XOH
60CC: 4B 115 HPOS PHA
60CD: 29 C0 116 AND #0C0
60CF: B5 00 117 STA HBASL
60D1: 4A 118 LSR A
60D2: 4A 119 LSR A
60D3: 05 00 120 ORA HBASL
60D5: B5 00 121 STA HBASL
60D7: 6B 122 PLA
60D8: B5 01 123 STA HBASH
60DA: 0A 124 ASL A
60DB: 0A 125 ASL A
60DC: 0A 126 ASL A
60DD: 26 01 127 ROL HBASH
60DE: 0A 128 ASL A
60E0: 26 01 129 ROL HBASH
60E2: 0A 130 ASL A
60E3: 66 00 131 ROR HBASL
60E5: A5 01 132 LDA HBASH
60E7: 29 1F 133 AND #01F
60E9: 09 40 134 ORA #040
60EB: B5 01 135 STA HBASH
60ED: BA 136 XPOS TXA
60EE: C0 00 137 CPY #00
60F0: F0 05 138 BEQ HPOSN2
60F2: A0 23 139 LDY #023
60FA: 69 04 140 ADC #04
60F6: CB 141 HPOSN1 INY

```

THIS ENTRY ONLY
COMPUTES Y & HMASK

60F7	E9 07	142	HPOSN2	SBC #57	
60F9	80 FB	143	DCS HPOSN1		
60FB	AA	144	TAX		
60FC	8D 4F 60	145	LDA MSKTBL-249, X		
60FF	85 02	146	STA HMASK		
6101	60	147	RTS		
6102		148	*		
6102		149	*CLER CLEAR BACK FLG+1 BITS		
6102		150	*		
6102	38	151	CLER	SEC	FIND XO-FLG
6103	AD FF 02	152	LDA XOL		DOUBLE BYTE
6106	ED FC 02	153	SBC FLGL		
6109	8D FC 02	154	STA XL		
610C	AD FE 02	155	LDA XOH		
610F	ED FB 02	156	SBC FLGH		
6112	8D FB 02	157	STA XH		WITH RESULT IN XL, XH
6115	AE FC 02	158	LDX XL		SETUP FOR XPOSN
6118	AC FB 02	159	LDY XH		
611B	20 ED 60	160	JSR XPOS		
611E	A5 02	161	LDA HMASK		DELETE THE BIT
6120	49 FF	162	EOR #5FF		@ Y, HMASK
6122	31 00	163	AND (HBASL), Y		ON LINE HBASL
6124	91 00	164	STA (HBASL), Y		
6126	18	165	CLC		
6127	AD FC 02	166	LDA XL		FAKE OUT TEST SO
612A	69 01	167	ADC #501		THAT
612C	CD FF 02	168	CMP XOL		WE ONLY GO TO
612F	AD FB 02	169	LDA XH		XD-1
6132	ED FE 02	170	SBC XOH		
6135	EE FC 02	171	INC XL		
6138	D0 03	172	BNE XOK		NOW INC XL, XH
613A	EE FB 02	173	INC XH		
613D	90 D6	174	XOK		CONTINUE
613F	A9 00	175	LDA #500		DONE CLEAR
6141	8D FC 02	176	STA FLGL		FLAG
6144	8D FB 02	177	STA FLGH		
6147	60	178	RTS		AND RETURN
6148		179	*		
6148		180	*MSKTBL HMASK LOOKUP TABLE		
6148		181	*		
6148	01 02 04				
614B	08	182	MSKTBL	DFB #01, #02, #04, #08	
614C	10 20 40	183		DFB #10, #20, #40	
614F		184		PAGE	
614F		185	*		
614F		186	*HIRES PICK STUPID PICK ROUTINE		
614F		187	*		
614F		188	*ALTM = #80 FOR COLOR SET 1		
614F		189	*ALTM = #00 FOR COLOR SET 2		
614F		190	*COLR = #AA FOR PUR&BLU		
614F		191	*COLR = #05 FOR GRN&ORG		
614F		192	*		
614F		193	ALTM	EGU #2FE	
614F		194	COLR	EGU #2FF	
614F		195	A1L	EGU #00	
614F		196	A1H	EGU #01	
614F		197	*		
614F		198	*HIRES PICK OFF		
614F		199	*		
614F	98	200	HPICK	TYA	SAVE Y, X AND ZPG
6150	48	201	PHA		
6151	A5 00	202	LDA A1L		#00
6153	48	203	PHA		
6154	A5 01	204	LDA A1H		#01
6156	48	205	PHA		
6157	A0 00	206	LDY #500		SET INDEX
6159	84 00	207	STY A1L		SET ZPG CTRS
615B	A9 40	208	LDA #540		
615D	85 01	209	STA A1H		TO BEGINNING OF
615F	81 00	210	LOOP	LDA (A1L), Y	PICK UP BYTE
6161	40 FE 02	211	EOR ALTM		CHANGE SETBIT
6164	30 04	212	BMI OKBYT		IF RIGHT SET

6166	A9 00	213	LDA #00		WRONG SET CLR BYT
6168	FD 0C	214	BEQ STOR1		ALWAYS TAKEN
616A	A5 00	215	LDA A1L		ARE ON ODD OR EVEN
616C	4A	216	LSR A		BYTE? THE CARRY KNOWS
616D	AD FF 02	217	LDA COLR		IF EVEN THEN
6170	80 02	218	BCS STOR		SHIFT ELSE STOR
6172	49 7F	219	EOR #57F		SHIFT MASK
6174	31 00	220	AND (A1L), Y		WIPE OUT EXTRA
6176	91 00	221	STA (A1L), Y		AND STORE IT
6178	A5 00	222	LDA A1L		GOTO NEXT BYTE
617A	C9 FF	223	CMP #5FF		WITH TEST FOR
617C	A5 01	224	LDA A1H		END OF HSCRN
617E	E9 5F	225	SBC #55F		(#5FFF)
6180	E6 00	226	INC A1L		
6182	D0 02	227	BNE CHLOP		
6184	E6 01	228	INC A1H		
6186	90 D7	229	BCC CHLOP		CC ON NOT END
6188	68	230	PLA		RETURN Y AND ZPG
6189	85 01	231	STA A1H		
618B	68	232	PLA		
618C	85 00	233	STA A1L		
618E	68	234	PLA		
618F	A8	235	TAY		
6190	60	236	RTS		
6191		237	PAGE		

*** SUCCESSFUL ASSEMBLY: NO ERRORS



NOTES AND CORRESPONDENCE





10260 Bandley Drive
Cupertino, California 95014
408 996-1010